FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Eyes in the sky: multi-drones surveillance technology

Francisco David Pereira da Silva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: António Pedro Rodrigues Aguiar

July 23, 2017

Resumo

Nos atuais sistemas autónomos, a utilização de equipas de drones é uma realidade emergente. Os drones são considerados uma tecnologia disruptiva que veio proporcionar soluções para diversos problemas. Um dos seus principais usos é no campo da vigilância.

Vigiar, por definição, é o ato de percorrer, cobrir, patrulhar e monitorizar uma certa área. Assim, vigiar uma área usando uma equipa de drones é um problema de cobertura como também um problema de gestão da cobertura da área. As estratégias utilizadas para tal, dependem diretamente do número de drones e da área a ser vigiada. Surgindo, essencialmente, duas situações.

Primeira, quando o número de drones não é suficiente para fazer uma cobertura total do espaço, sendo necessário criar trajetos que façam os drones patrulhar toda a área, uniforme e periodicamente.

Segunda, quando o número de drones é suficiente para fazer uma cobertura total da área, e então, precisamos de a monitorizar em tempo-real. Contudo, a questão para a cobertura de uma área usando vários drones é de como vamos distribuir as tarefas por cada agente por forma a termos a solução ideal.

Tendo isso em conta, o foco desta dissertação é na resolução dessas situações de vigilância de uma área através do uso de um conjunto de drones.

Para a primeira situação, um algoritmo baseado na vigilância não comunicativa é apresentado. Nesse mesmo método, a área a ser vigiada é um polígono convexo que é patrulhado através de uma serie de movimentos de varredura com drones equipados com câmaras. A direção da varredura é aquela que minimiza o número de curvas do percurso. Depois, as extremidades desses mesmos caminhos, são modeladas como nós de um gráfico, transformando o problema de atribuição dos caminhos para cada drone num problema de Caixeiro Viajante onde, a partir da minimização do valor máximo do sistema, obtemos os caminhos ideais para cada drone percorrer a área a ser vigiada.

Para a segunda situação, um algoritmo comunicativo é usado e apresentado. Neste algoritmo, é assumido que os drones tem câmaras apontadas para o solo e que quando mais alto se encontram, mais pobre é a qualidade da imagem da área vigiada. Nesse sentido, o algoritmo fornece a melhor configuração para os drones se distribuírem na área a ser monitorizada através da maximização de um conjunto de critérios de qualidade de imagem ajustando as suas posições através de leis distribuídas de controlo.

Ambas metodologias são detalhadamente expostas e validadas através de um conjunto de simulações que serão analizadas.

Abstract

The use of multi-drones teams is an emergent reality of the autonomous systems. Drones are a disruptive technology that came to give solution to many problems, finding in them, a wide range of applications. One of their main applications is surveillance.

To surveil an area is by definition the act of cover, patrol and monitor a certain area. Thus, to surveil an area with a team of drones is not only a coverage problem but also a task assignment problem that relies on the number of drones and area to be covered itself. This way two situations will emerge.

Firstly, when the drones are not enough to cover all the area. In this case, the area coverage needs to be patrolled periodically.

Secondly, when there are enough drones to cover all the area. In this case, the area needs to be real-time monitored. However, the main question lies in how we are going to disjointly assign the tasks so that the system performance will be optimal.

This dissertation focus on the resolution of those two area coverage problems, associated with multi drones surveillance.

For the first problem a non-communicative algorithm is presented and used. In the non-communicative method, the area to be surveilled is a convex polygon covered by consecutive back and forth motions of drones equipped with cameras. The drones sweeping direction is such that path turns are minimized. Those paths extremes are then modeled as nodes of a graph, transforming the task assignment problem into a *Traveler Salesmen Person* problem, where through the minimization of the maximum value, we achieve the best coverage solution for a certain convex polygon area.

For the second problem a communicative algorithm is presented and used. In this algorithm it is assumed that drones have cameras facing downwards and that the higher their altitudes the poorest is the image quality. This way this algorithm gives to the monitoring area the best configuration by maximizing a joint coverage quality criterion and adjusting the drones location through the use of distributed control laws.

Both algorithm's methodology are detailedly exposed and validated by a set of different simulation posteriorly analyzed.

Acknowledgements

Once this dissertation required a lot of technical knowledge and emotional management, I would like to divide my compliments in two distinct parts.

On the first part, I would like to express my gratitude to my supervisor António Pedro Rodrigues Aguiar for his tremendous patience, guidance and for introducing me the topic.

I would like to compliment Andrea Alessandretti for his availability to answer technical questions and helping with dissertation development. Maria João Correia Soares da Costa for her LSTS's Toolchain tutorials and further support. Also, would like to thank António Sérgio Borges dos Santos Ferreira for his readiness on pointing the best person to deal with my issue. Still about the LSTS's people, I would like to laud José Pinto and Tiago Marques for their trust and optimistic preview for the SITL simulations and also for their effort to make compatible the Matlab algorithms with LSTS toolchain.

For last, and a bit more impersonal, would like to salute Gustavo Avellar for providing his master thesis codes and documentation which held one of my dissertation parts. And to salute Sotiris Papatheodorus for give away his code for deviation and his knowledge about cooperative visual area coverage algorithms.

On the second part of this acknowledgements section, I would like to greet my girlfriend Rita Ferreira for pressuring me every mornings to wake up. For not allowing me to do anything but work hard, so I could satisfy the deadlines imposed. For listening me when I needed to grizzle, for giving her opinion, for giving me confidence to carry on. For understanding my bad mood on last couple weeks/months. This is also her work, and I would like to emphasize that. Without her, this work was not possible.

Also like to thank my family, this is, my parents and brothers. For paying the bills and for giving me love, education, food and shelter.

Last but not least, I would like to compliment my friends, colleagues, brothers-in-arms, whatever you call them. This paragraph had to be here because I need to greet their efforts, their friendship and their wisdom. They made me a lot more than I was before I met them. I could specify their names but I am sure that is not relevant. If any of them is reading this I am sure they will know this is for them. My thanks to you guys.

As final note, would like to refer all the persons and situations who stumble across my path during these years. It wasn't always easy, there was good times but also bad times. Sometimes I was full of energy others I was tired. But, I never gave up. I always moved forward to face the circumstances and learn from them.

Hope the bonds made along these years do not weaken.

Francisco Silva

"To act – that is true wisdom.
I can be what I want to be,
but I have to want whatever it is.
Success consists in being successful,
not in having the potential for success.
Any wide piece of ground is
the potential site of a palace,
but there's no palace until it's built."

Fernando Pessoa in *The Book of Disquiet*

Contents

List of Figures

List of Tables

Abbreviations and Symbols

AGL	Height above ground level
BLOS	Beyond line of sight
ALFUS	Autonomy Levels for Unmanned Systems
API	Application Programming Interface
ATC	Air Traffic Control
C++	Object Oriented Programming Language
C2	Command and Control
C3	Communication Command and Control
CDL	Common Data Link
DoD	Department of Defense
DSSA	Distributed Self-Spreading Algorithm
DTN	Disruptive Tolerant Networks
FAA	Federal Aviation Administration
FOV	Field of View
FWUAV	Fixed-Wing Unmanned Aerial Vehicle
GCS	Ground Control System
GNC	Guidance Navigation and Control
GPS	Global Positioning System
GS	Ground Station
HALE	High Altitude Low Endurance
HITL	Software-in-the-loop
HRI	Human-Robot Interaction
HTOL	Horizontal Take-Off & Landing
IMU	Inertial Measurement Unit
LALE	Low Altitude Long Endurance
LARYNX	Long ranged gun with Lynx engine
LASE	Low Altitude Short Endurance
LOS	Line of sight
LSTS	Laboratório de Sistemas e Tecnologia Subaquática
MAA	Multi Aerial Vehicle
MALE	Medium Altitude Low Endurance
MANET	Mobile Ad-hoc Networks
MAV	Micro Air Vehicle
MRMP	Multi-Robot Motion Planning
MRS	Multi-Robot System
NAV	Nano Air Vehicle

NP	Non-deterministic Polynomial time
OODA	Observe, Orient, Decide and Act cycle
OS	Operative System
RC	Radio Controlled
RF	Radio Frequency
RPA	Remotely Piloted Aircraft
RPM	Rotation per minute
RPV	Remote Piloted Vehicle
SATCOM	Satellite Communications
SD	Smart Dust
SITL	Software-in-the-Loop
TSP	Traveler Sales Person
UA	Unmanned Aircraft
UAS	Unmanned Aircraft Systems
UAV	Unmanned Aerial Vehicle
UAVS	Unmanned Aerial Vehicle System
UK	United Kingdom
UMS	Unmanned System
USA	United States of America
VDDA	VD-based Deployment Algorithm
VTOL	Vertical Take-Off & Landing

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAV) are remotely operated aircraft systems with no pilot on board. Also called drones, this technology has urged with the need of better suited solutions for dull, dirty or dangerous missions apart from manned aircrafts. Nowadays, drones are an emerging disruptive technology that each day is drawing more attention into many different fields of application. The increasing evolution of materials, computer intelligence and robotics created hundreds of different drones configurations. They can be big and expensive drones, used in warfare but they also have light, robust and cheap drones on the market for commercial uses.

The inherits tactical advantage of having literally "eyes in the sky", made of drones an huge success for applications when vision is the biggest issue. Allied with the fact that they are smart machines with big data processing power and quick response, make of a well designed drone better than any pilot or human for some specific tasks.

Drones have today many applications in a wide set of fields: they are used by armies to patrol the borders, to execute search and destroy mission or simply for surveillance. On the other hand, there is the commercial field, where drones are used for search and rescue missions, wildfire detection, deliveries, film-making, crops control, monitoring, sampling, exploring or just algorithms research.

Due to the dependence created around these robots, the scientific community is struggling to find ways to respond to all of these applications constant requirements. One of the most common issues is the area coverage once that, has mentioned before, one of the biggest advantages of the drones is to have "eyes in the sky". For that we need to develop a set of technologies like communication protocols, image recognition, flight control and other autonomous tasks.

Despite drones are used everywhere, they still have a big bottleneck which is an average low endurance. One of the main solutions, is to use a team of drones, decreasing the energy consumption of each agent by splitting the job among all drones.

In this dissertation, we scope the common problem of barely all of these applications which is the surveillance. Thus, in this work we will study the literature, adopt the works made in the field of surveillance using multi-drones, and will explore and deceive strategies and methods for solving the problem of area coverage.

1.1 Context and Goals

In the field of surveillance, drones have taken the bigger part of the tasks. By using multi-drones systems we can reduce the impact of drones low endurance and enhance big applications by distributing the work among several drones.

Patrolling, in a more general sense, may involve the task of visiting selected places in a given area in order to assess the state of the environment in relation to the presence of intruders or the occurrence of unforeseen events. It is therefore, the act of search an area, eventually in regular intervals, in order to supervise, cover or protect that area.

The area coverage problem is one of the classic problems in the field of surveillance and patrolling. It is essentially the search of the best way of patrolling a certain area covered by a given number of drones. If the entire terrain cannot be monitored at all times, each location in the target area is periodically monitored.

However the surveillance task, using a multi team of drones poses several challenges because not only one has to address the design of how the drones will cover the area, but which area can they cover, how its tasks are disjointly assigned to the drones and what's the best solution for that problem. Involving several different problems like communication, flight control and coverage strategy.

In this dissertation, the solution proposed is divided into two parts: in the first part, we deal with an area coverage problem situation where we have shortened number of drones, which mean we cannot use a real time monitoring system to cover all the area. This way we want to develop a method where each location in the target area is periodically monitored.

In the second part, we study the coverage problem situation where we have enough number of drones to cover all the area efficiently. This way we propose a method to real-time monitor all area.

1.2 Motivation

The motivation for this work comes from the need of better solutions for the existing area coverage problems. Most of the solutions used so far are outdated and do not incorporate the new features of the drones, which can be more autonomous, quicker and effective. The methods proposed in this dissertation are dynamic and give optimal solutions for area coverage problems where in first place, we want to patrol and surveil an area most periodically as possible. And in second place, when we want to real time monitor an area.

As aforementioned, drones are the new disruptive technology that will find a place to act in every field. The fields where they are more needed is on the visual coverage problems, like wildfire sensing, sampling, monitoring and exploring.

When we have that much applications, which are very important but rely on so outdated, low intelligence methodologies. There's the need to improve, so these applications get also improved.

1.3 Contribution

This dissertation contributes to the area of surveillance using multiple drones by exploring two different methods to solve the problems related with area coverage when the number of drones is limited. In the first problem, to optimize the periodicity of the coverage, we want to minimize the time to cover it.

We present a solution for the problem of minimum time coverage of ground areas using a group of drones equipped with cameras facing downwards. The solution given is composed by several steps. In the first step, an area partitioning accordingly to [17] and [23], minimizes the number of turns, leading consequently to faster and higher endurance paths. Then we plot rows in the direction of minimization of turns, to perform the back and forth motion coverage. To give the optimized solution, we model the problem as a graph and solve it like a *Traveler Salesmen Person* (TSP) problem. From there we obtain a group of routes that represents the minimization of the maximum time taken to perform the coverage of the area. We exploit also the results in [18], where they consider in the problem methodology realistic constraints such as the number of operators and the drone setup time. Also, the optimal solution refers if all the available drones are necessary to execute the mission or not. The mission tasks assignments is done accordingly to the solution given by the optimization of the objective function.

To solve the problem of real time monitoring all area, we need to have a large number of drones, enough to cover the area. In the adopted solution based in [16], each drone is assumed to be with cameras facing downwards. It is also considered that all the region sensed by that camera corresponds to an image that has higher or lower quality depending on the drone altitude. When the drone is at a high altitude the image quality is poor but the surveilled area is wide, in the opposite, the image quality is good but the covered area is small. Thus being the altitude proportional to the area covered whereas the quality of the covered area decreases with the altitude. In this method, a distributed control law is designed through the movement of drones among the area in order to maximize a joint coverage quality criterion where area covered and area covered quality are the main focus [19].

1.4 Structure

This dissertation is composed by seven Chapters. Chapter 1 describes the introduction and provide the context, motivation and contributions outline. In Chapter 2 several key aspects of Unmanned Aerial System (UAS) are explained and all other data related to drones somehow is exposed for extra knowledge on the subject. In Chapter 3 the literature of the subject is reviewed and related works briefly analyzed. The junction of last 2 Chapters compose the state of art. In Chapter 4 the problems stated in the introduction are formulated and assumptions for each case are outlined. Further, in the Chapter 5 the area coverage problem assumptions made are taken into account to formulate the methodologies to solve those problems. Then, in Chapter 6 simulations are presented to validate the implementation of the methodologies presented in Chapter 5 and results obtained are analyzed and commented. Finally, in Chapter 7 a last insight of the dissertation conclusions are made. In there, we briefly explain the work conclusions, the satisfaction of the initial proposed work and finally the future work recommendations.

Chapter 2

Background

Unmanned Aircraft Systems (UAS), also commonly named as drone, is a disruptive technology with tremendous potential to revolutionize warfare and to enable new civilian applications. The emerging possible applications coming from this technology lead into many studies and researches on the area.

As shown in [24], [20], [7], [5] and many other bibliographic references, UAS may vary in different ways according to the applications. The most basic is the flying device's shape [20] [7] [5].

In this chapter we will address UAS key concepts and technologies. The expert reader, comfortable with UAS terms, designations and their *Modus Operandi*, may go directly to chapter 3.

2.1 UAS

The term UAS was adopted by United States Department of Defense (DoD) and United States Federal Aviation Administration (FAA), accordingly to [5]. Further on, the term UAS was also adopted by other agencies around the globe [25]. This term, comparatively to UAV, gives a more wide view of what this technology is composed by because it gives importance not only to the flight device, the UAV, but also to the other infrastructure which UAVs could not work without. Other names are used to describe this technology however they usually do not include all the systems components [20].

Unmanned Aircraft System (UAS) is defined as a system whose components include the aerial vehicles, called Unmanned Aerial Vehicle (UAV), and the associated equipment that do not carry an operator. This is, the system that includes the aerial vehicles, the autonomous flight controller, the communication system, the ground control station, the sensors, actuators and all other peripheral components. In Fig. 2.1 you can see a simple scheme of an UAS basic architecture. Although the system architecture seems very simple at first sight, the interaction between those different structures it is very complex and may change a lot depending on the UAS purpose. In Fig. 2.2 you can see an example of how complex the architecture might become.

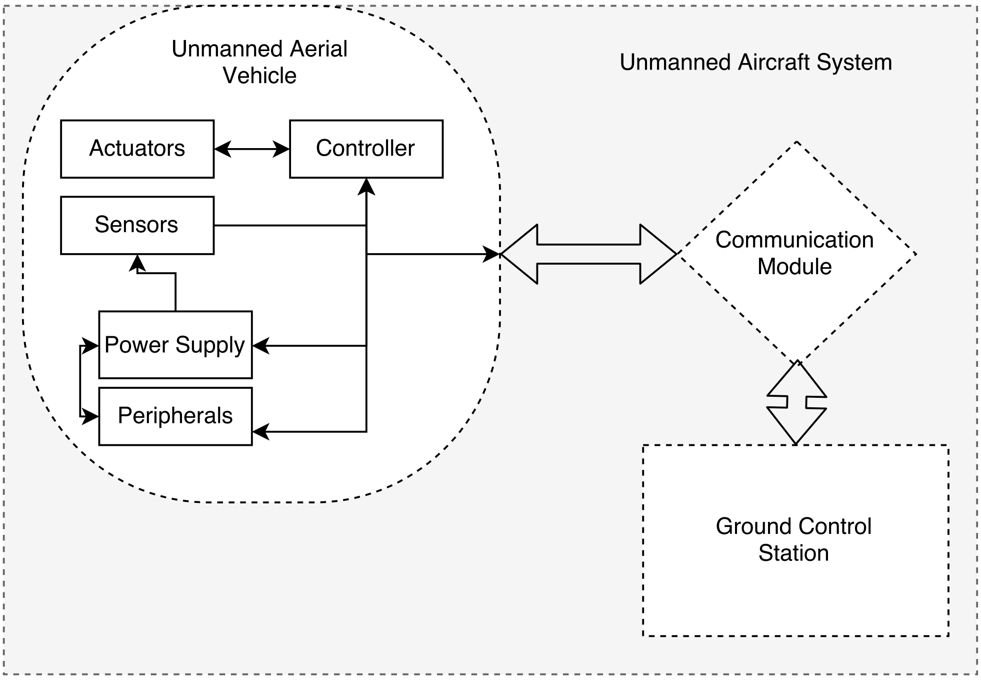


Figure 2.1: Basic architecture of an UAS [1].

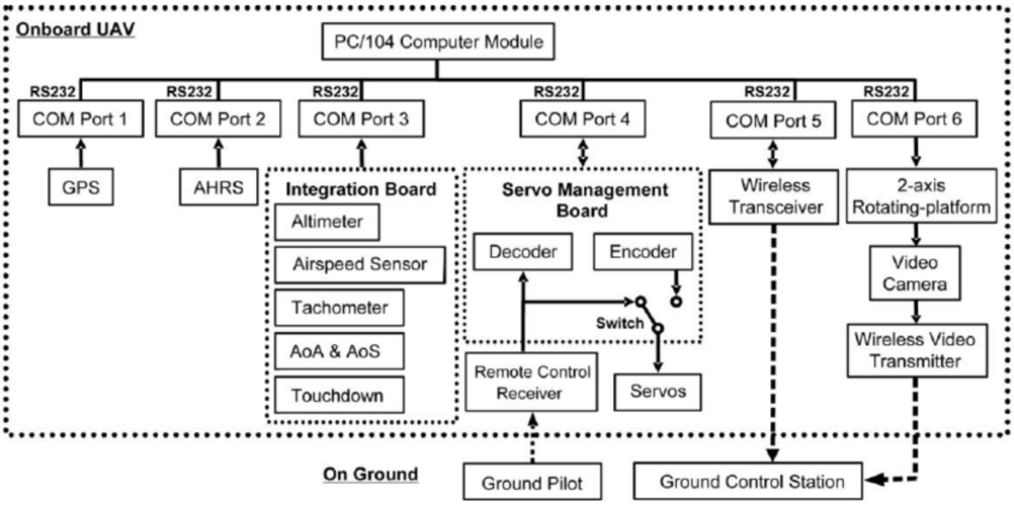


Figure 2.2: Architecture of an on board UAS module [2].

In Fig. 2.1 we can clearly see the blocks that compose an UAS. The UAV blocks can be generally defined as the controller, actuators, sensors, power supply and peripherals. The controller block is responsible for receiving the data sent by the Ground Control System (GCS) in case of operator's intervention, or in case of autonomous flight, for receiving the data from the Inertial Measurement Unit (IMU), Altitude sensor, Wind speed sensor, GPS, among others. Interpret it, and send commands to the actuators. The sensor block is responsible for collecting the highest amount of information possible. The actuators are generally the thruster or propellers of the UAV, however they may include another features. Power Supply is responsible for power supply all the components on the aircraft. Peripherals are all the extra hardware components that are not essential to the UAV autonomous flight. However they may be crucial to successfully accomplish some missions. Communication Module as it suggest, is the layer responsible for exchange information between the GCS and the UAV. As last, we have the GCS which is where the operator can plan the mission, read UAV data and send commands. In case of loss of control, the operator can also take command of the UAV and fly it remotely [20]. In Fig. 2.2 we can see a more detailed UAS system architecture that show the increase of complexity when we get an insight in one of the blocks referred before.

UAS can be divided into these three big blocks. In each one of these blocks there are types, characteristics and different classifications. In the further sections we will explore deeper how each block of this system works and what characteristics do they have.

2.2 History

Since dawn of time, the mankind as dreamed to conquer the skies. As result, the human developed the aircraft. The urgent need of better aircrafts for many uses alongside with technology development resulted in the birth of UAV.

The first record of what can be called an UAS was with warfare intentions when in August, 1849 the Austrians used hot air balloons loaded with explosives controlled by timed fuses to fly over the sky of Venice [26]. The damage done by these balloons was weak because some balloons did not went into the right direction. Despite the calculations for the launching positions, at the time, there was no way to fight against wind direction changes and so the balloon's effect was diminished. However, this was very scary at the time because nothing like that was used before. Although is primitive this was the first unmanned aerial vehicle [27].

Later on, in 1916 the Americans Lawrence and Sperry manufactured the first UAV using their developed gyroscope to stabilize the body in an attempt of creating an "autopilot". This is known as the beginning of the "attitude control" which came to be used on aircraft's automatic steering [26]. During the World War I, Dayton-Wright Airplane Company invented pilotless torpedo [28]. He relied on mechanical gyroscopes to maintain straight level flight, and flied until ran out of fuel. A catapult was needed to launch this torpedo. This torpedo was baptized as "Kettering Bug" in honor to one of his creators. It was a small biplane powered by a De Palma four cylinder engine. Using a wooden framework, which was covered with pasteboard on the fuselage and tail surfaces, and with doped paper and muslin on the wings. The wings had 10^o of dihedral, which was intended to improve stability on take-off. Flight distance was measured by counting air-impeller revolutions, and when the preset distance was reached, the engine was cut, the wings detached, and the Bug fell to the ground. It was capable of flight to distances over the 80 km with payloads of approximately 90 kg. Despite this "guided torpedo" was never used in battlefield he was a plot twist in UAV research and development history. During the war, USA was reluctant to enter in this concept of "flying bombs". However, with the "Kettering Bug" demonstrations, USA joined the research and officially baptized this torpedo as "Liberty Eagle" [29] [30]. When the war was over, this project was shutdown because improved gyro stabilizers and radio-controls systems started to surge. In the Figs. 2.3 and 2.4 you can see the Austrian air balloon and the famous "Kettering Bug" also known as "Liberty Eagle", respectively.

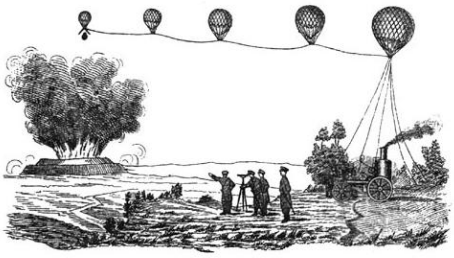


Figure 2.3: Hot air balloons Venice Attack. August, 1849. From *Monash University*.

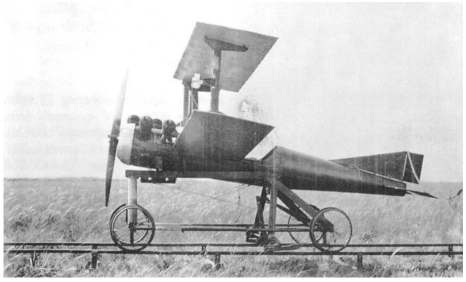


Figure 2.4: "Kettering Bug" also known as "Liberty Eagle". October, 1918. From *Monash University*.

The following years, during the inter war period, there was a bunch of different projects. One of them was the Long ranged Gun with Lynx Engine (LARYNX), a cruise missile in the form of a monoplane aircraft that could be launched from a warship and fly under autopilot. This aircraft used radio knowledge to be remotely controlled and hit the targets with better precision than ever before. At that time, was such a success that radio controlled (RC) aircrafts started being mass studied and produced. From this run to RC aircrafts, emerged the well known "DH.82B Queen Bee". Developed by United Kingdom (UK), derived from Havilland Tiger moth. This aircraft was the first fully autonomous operated vehicle. Cockpit position was equipped with radio-control gear including pneumatically-operated servo units linked to the aircraft rudder and elevator controls. With a flight range of 480 km, this aircraft was the first called a 'drone'. Also nominated nowadays as the "mother of drones" [3] [31]. In Fig. 2.5 we can see an example of the aircraft previously referred.

Meanwhile, Reginald Denny an actor and airplane enthusiast created a company where he made use of Edward M. Sorensen recent achievements on radio-controller aircrafts (US Patent 2,490,844 filed in May of 1940; Patent 2,408,819 filed May 16, 1940 and patent 2,482,804 filed May 16, 1940).

For first time in the UAV history the aircraft could be fully operated from a ground station. The operator could know what the plane was doing, such as climbing, altitude, banking, direction, RPM and other instrumentation. Without these patents the early radio-controlled aircraft could only operate within visual sight of the ground pilot.

These aircrafts were such a success that in 1940, with the need of better weapons, the US army made a huge contract with Reginald's Company. The purpose was to mass-produce these RC aircrafts to use them in war and training simulations [4]. The name of this RC aircraft was "Radioplane OQ-2" and an example can be seen in Fig. 2.6. Despite the further variations the main principle was the same.

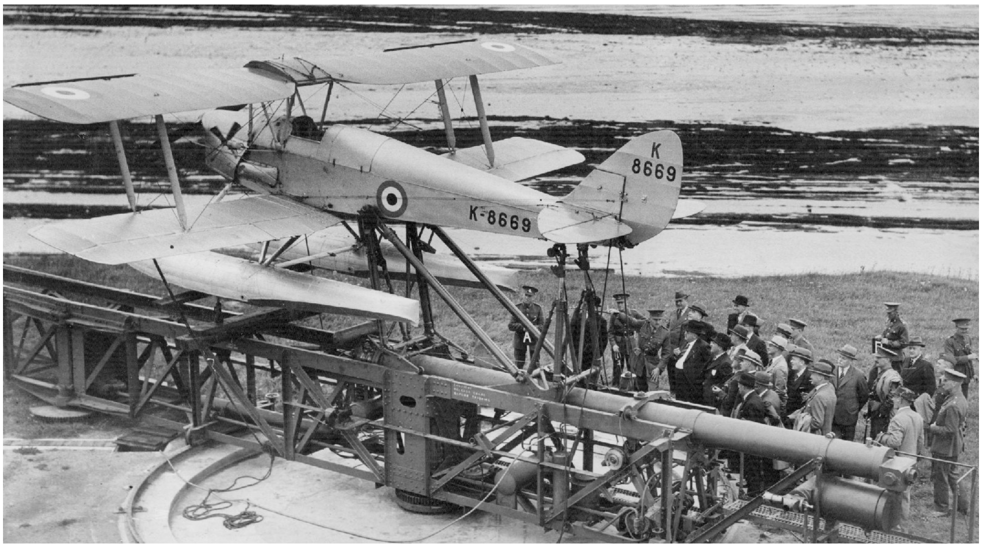


Figure 2.5: DH82b Queen Bee (K8669) preparing for launch, April 1942 [3].

During the war the pulsejet technology emerged which meant the drones and torpedoes were now able to travel further distances, reach new altitudes and travel faster than ever before.

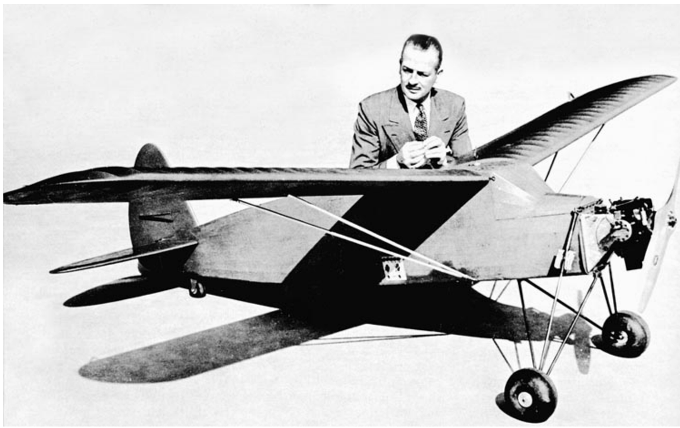


Figure 2.6: Reginald Debby and his "Radioplane OQ-2", 1940 [4].

Despite the evolution made on the area in those years, the true development of UAV as we know today happened on the last 50 years.

After the World War II, great scientific discoveries were made due to the Cold War. This race for global domain pressured the nations and scientific community to acquire constantly better weapons and better technologies. Those improvements turned drones into complete warfare machines essential to military scenario. They were capable of patrolling an area, realize search and destroy missions, explore areas and collect data. In Fig. 2.7 we can see the first jet-powered UAV, built by the USA in 1951.

Some major history events like the nuclear programs, Vietnam war and Yom Kippur war, used

Figure 2.7: "Ryan Firebee" the first jet-powered UAV at Muzeyon Heyl ha-Avir, Hatzerim airbase, Israel. 2006. From *wikipedia*.

recent developed drone technologies like monitoring and area patrolling platforms. In Yom Kippur War, soviet-supplied surface-to-air missile batteries in Egypt and Syria caused heavy damage to Israeli Jet fighters. As result, Israel developed the first real-time UAS surveillance [32]. This is pretty interesting in the sequence of this dissertation once it was here where recent real-time surveillance systems with drones were born. Israel with the conflict, began to develop smaller and cheaper UAV. These were small aircraft that adopted small engines. They carried video cameras and transmitted images to the operator's location.

With the increase of processing power, development of better networks, increasingly Artificial Intelligence and controllers capabilities. Drones found their use not only in warfare but also in civil usage. The drones from modern era can be flying killing machines or can be used for the good. With the massification of drones usage came the need to create protocols to regularize their usage and limitations. In order to do that, drones had to be differentiated. As result, drones can be classified and divided with different classes derived from their characteristics or simply usage [20]. In Fig. 2.8, 2.9, 2.10 and 2.11 we can see cutting-edge modern era drones examples in many different fields. Some of them are, surveillance, exploration, transportation, recreational and many others.

Figure 2.8: General Atomics MQ-9 Reaper an US military drone [5].

Figure 2.9: "DJI Phantom 3" a civil recreational quadrocopter drone [6].

Figure 2.10: Amazon air prime 2 an octocopter used in deliveries [6].

Figure 2.11: NASA's *Helios* a solar powered drone prototype. From *DARPA*.

Interesting bibliography on this matter is [33], [34], [35] and [36].

2.3 UAV

UAVs are aircraft with no on-board crew or passengers. They appeared with the need of sending aircrafts to places where the risks were too high or impractical to reach by manned aircrafts. They can be autonomous (drones) or remotely piloted vehicles (RPVs). They can fly for long periods of time at a controlled level of speed and altitude and have a role in many aspects of aviation. Their characteristics and automation depend directly from the complexity of the task to be performed [20].

Thus, as they change alongside with the function to be developed, drones needed to be classified. Since drones are extremely versatile also are their configurations. In that way, drones classification has many ways to be done. It can be by their autonomous levels, range and endurance, components, weight or simply shape and size [7].

2.4 Classification

Dedicated control systems may be devoted to large UAVs, mounted aboard vehicles or in trailers, to enable close proximity to UAVs limited by range or communication capabilities. The smallest categories of UAVs often are accompanied by ground-control stations consisting of laptop computers and other components small enough to be carried easily with the aircraft in small vehicles, aboard boats, or in backpacks.

In [20] drone classification has been done accordingly to drones category, UAV's weight, operating range and other characteristics as shown in Table 2.1. Where AGL stand for, Height above ground level, LOS stands for Line of sight and BLOS for Beyond Line of Sight.

The work in [7] describes a variety of UAS platforms where the advantages of each is related to the demands of users in the scientific research sector. They classified the drones platforms in civil, scientific and military uses. In their drones classifications, they classified them as MAVs (Micro or Miniature Air Vehicles), NAVs (Nano Air Vehicles), VTOL (Vertical Take-Off & Landing), LASE (Low Altitude, Short-Endurance), LASE Close, LALE (Low Altitude, Long Endurance), MALE

Table 2.1: Classification of UAVS based on [20].

Category	UAV's Weight	Normal Operating Altitude	Radius of Mission	Endurance	Altitude
MICRO	<2 kg	Up to 60m AGL	5 km (LOS)	A few hours	Very low Altitude
MINI	2-20 kg	Up to 900m AGL	25 km (LOS)	Up to 2 days	Low altitude
SMALL	20-150 kg	Up to 1500m AGL	50 km (LOS)	Up to 2 days	Low Altitude
TACTICAL	150-600 kg	Up to 3000m AGL	200 km (LOS)	Up to 2 days	Low Altitude
MALE	>600 kg	Up to 13700m AGL	Unlimited (BLOS)	Days/Weeks	Medium Altitude
HALE	>600 kg	Up to 19800m AGL	Unlimited (BLOS)	Days/Weeks	High Altitude
STRIKE/ COMBAT	>600 kg	Up to 19800m AGL	Unlimited (BLOS)	Days/Weeks	High Altitude

(Medium Altitude, Long Endurance), and HALE (High Altitude, Long Endurance). In Fig. 2.12 we see one schematic representation of Drones platforms classification.

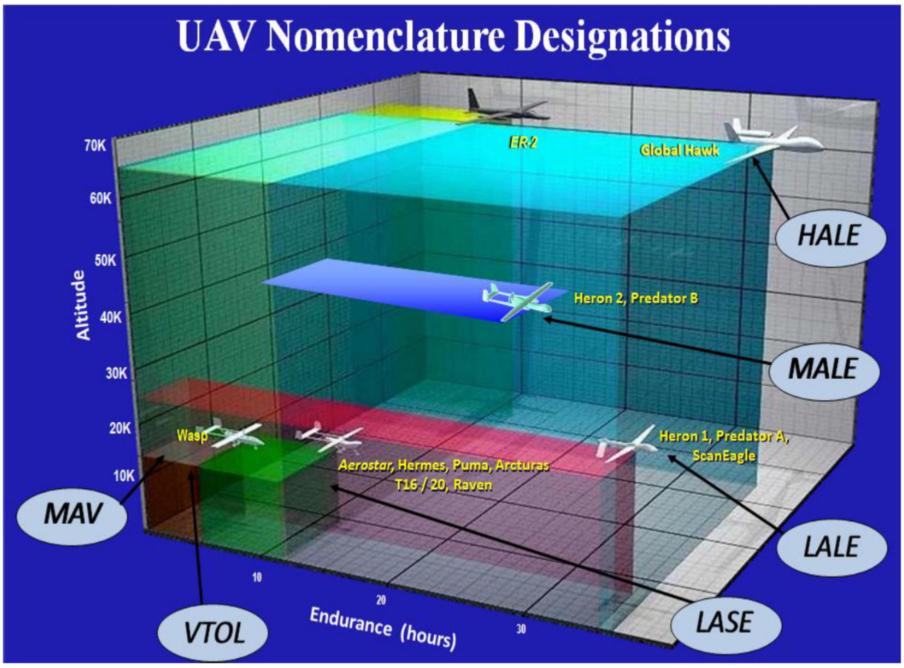


Figure 2.12: Classification of UAS platforms from [5]

In [37], the drones are categorized as three main types, namely, micro and mini UAVs, tactical UAVs, and strategic UAVs. The tactical UAVs are distributed into six subcategories: close range, short range, medium range, long range, endurance, and medium altitude long endurance (MALE) UAVs. See also [22] where drones were classified as micro, mini, tactical, medium and high altitude, and heavy types. Other classification and class division are presented in [38], [39], [6].

Those classification are mainly based on systems capabilities. Furthermore alternative classifications exist, based on their usage or functionality [6] [40] such as:

- Target and Decoy;
- Reconnaissance;
- Combat;
- Logistics;
- 3D Mapping;
- Agricultural;
- Disaster Relief;
- Research;
- Civil and commercial;
- Swarming.

Furthermore, different types of drones evolved from the advancement in miniaturization of electronic components, such as sensors, micro-processors, batteries, and navigation systems [41]. Drones size range is from vast fixed-wing unmanned air vehicle (FWUAV) to smart dust (SD), which consists of many tiny micro-electro-mechanical systems including sensors or robots. In Fig. 2.13 can be seen a spectrum of different type presented through their wingspan and weight.

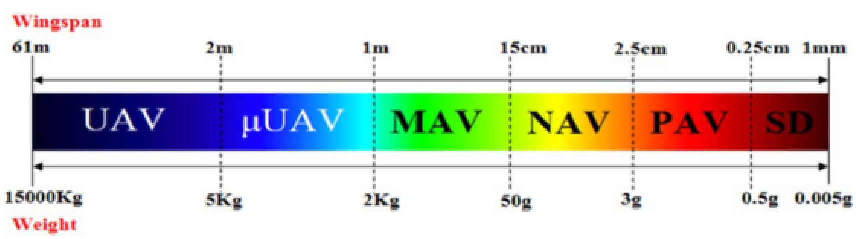


Figure 2.13: Spectrum of drones from UAV to SD [6].

The main aspects that distinguish UAVs from other types of small drones (such as MAVs and NAVs) include the operational purpose of the vehicle, the materials used in its fabrication, and the complexity and cost of the control system [42]. Different mission requirements created various types of UAVs. For this reason, it is also often useful to categorize UAVs in terms of their mission capabilities. In Fig. 2.14 an unconventional category of drones are listed. In there, all types of existent drones are displayed. Their classification mainly rely on their size class and their thruster type and position.

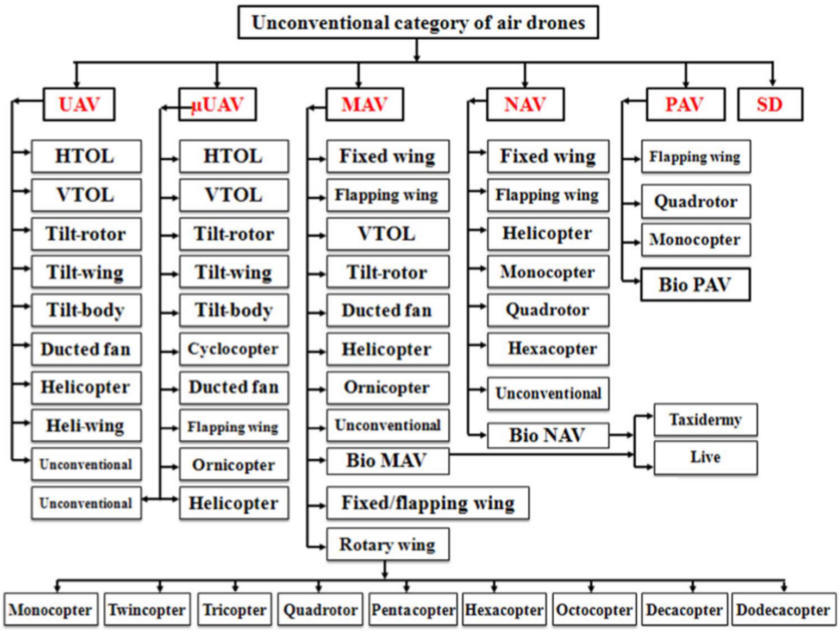


Figure 2.14: Types of drones [6].

2.5 Components

The unmanned aircrafts have similar components to manned aircrafts, but without the cockpit and all the pilot support systems. Even if the UAV carry payloads such as cameras or heavy weapons they will always be lighter than the comparable manned aircraft. This make of UAVs generally smaller. Compared to the manned aircrafts, UAVs replace the cockpit window for cameras and video link; physical controls for radio-transmitted digital plus a set of autonomous controllers and algorithms to control the flight. UAVs components perform generally the same task and use the same principles, however they can change in size, robustness, capability, response time and many other aspects.

As referred on section 2.1, the general system architecture its composed by two Hardware modules, the ground station and the vehicle. In the side of the vehicle we have the sensor to collect data, the actuator to perform the tasks and flight, the energy supply so we can power, the processor where the data is collected and processed and the communication module so the vehicle can send and receive information from the ground station like a new path to follow, a new tasks or even new flight plans. In the side of the ground module, we have the central computer, where the operator can control the UAV remotely, the outer loop control algorithms, the flight plans and the instrumentation, needed to support the communication with one or multiple drones [7].

Briefly explained on the following subsections will be the UAV modules main components and function.

2.5.1 Body

The body is the frame of the aerial vehicle where all the components are built. This is one of the most important components of the UAV once it is where all the on-board components will be assembled. There are many kinds of UA frames, and the UAV type is roughly defined by the frame. The body of the UA will define his aerodynamic configuration, size, location and type of propellers, weight, payloads, etc. Generally the, UAV's body fall into one of the following categories [5]: Fixed-wing UA, Rotary-wing UA, Blimps or Flapping-wing UA [20].

Despite in Fig. 2.14 we can see a lot of different drone types, most of them are derivations from these 4 categories.

Fixed-Wing UA are very similar to normal airplane, they usually travel further distances, higher speeds, are bigger but also expensive and more limited in terms of civil use, once they flight dynamics are less flexible when compared to the rotary-wings [6]. Rotary-wings are usually less robust than fixed-wing aircrafts however they are cheaper, easily maintained, easily controlled and much more flexible in terms of flight dynamics. There are many kinds of rotary-wing and the common name that is given to a specific model derives from the number of rotors. Blimps and Flapping-wings UA are less common however they have some advantages in certain fields of motion research. Blimps are usually such as balloons and are used, per example, for forecasting. Flapping-wings are normally flexible wings UAV used to simulate birds and flying insects [6] [20] [43]. The shape of the UAV body directly implies big changes in aerodynamics and flight controllers.

In Fig. 2.17 we see several examples of different configurations of fixed-wings UAV. Beside from the wing shape they also can vary in number of wings and propellers. In Fig. 2.18 different number and disposition of rotors is shown. The common designation of rotary-wing UA usually is the number of rotors plus the word "copter". For instance, a four rotors rotary-wing UA is commonly named as "quadrocopter". Also in Figs. 2.15 and 2.16 example of a blimp and a flapping-wing UA are shown, respectively.

2.5.2 Sensors and Actuators

Drones sensors and actuators can be divided into core and peripherals. The core sensors and actuators are those who the drone needs to fly and maintain his security. Peripherals are all the other extra sensor and actuators who allow drones to do their specific tasks.

The main actuators, also called outputs, include the digital electronic speed controllers linked to the engines and propellers.

Figure 2.15: US blimp used for patrolling [6].

Figure 2.16: SmartBird, a flapping-wing drone created by Markus Fischer that copies an eagle flying dynamics. From Festo.

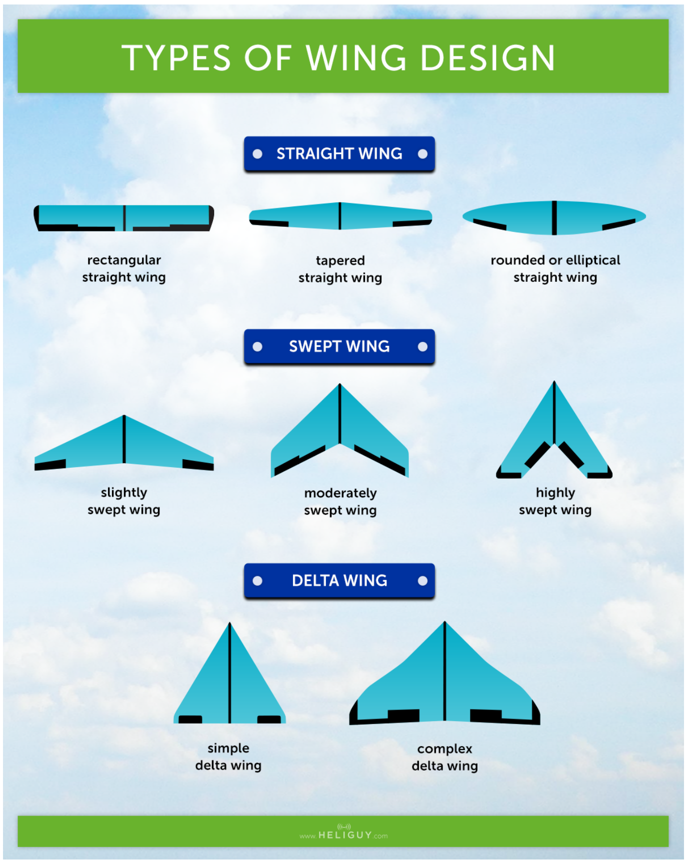


Figure 2.17: Fixed-wing UAV body's design examples from *"Heliguy"*.

The main sensors, also called inputs, are flight control sensors. Exteroceptive and expropri-oceptive sensors [42], collision avoidance sensors, compass, gyroscopes and accelerometers are part of an unit called IMU (Inertial Measurement Unit), localization sensors like for GPS and other telemetry, airspeed sensors and a barometer are some examples of main sensors present in drones. Off course many others sensors and actuators exist but those are not as essential for UAVs core mechanisms.

Figure 2.18: Rotary-wing UAV design examples from *"DesignBoom"*.

When these two components modules work together, high levels of autonomy can be achieved. The work and interaction developed between sensors and actuators can be compared with human body feedback mechanism [44]. The higher the amount of sensors and actuators the higher is the level of autonomy of the UAV. In Figs. 2.2, 2.19 and 2.20 examples of UAV sensors and actuators on board interaction are shown.

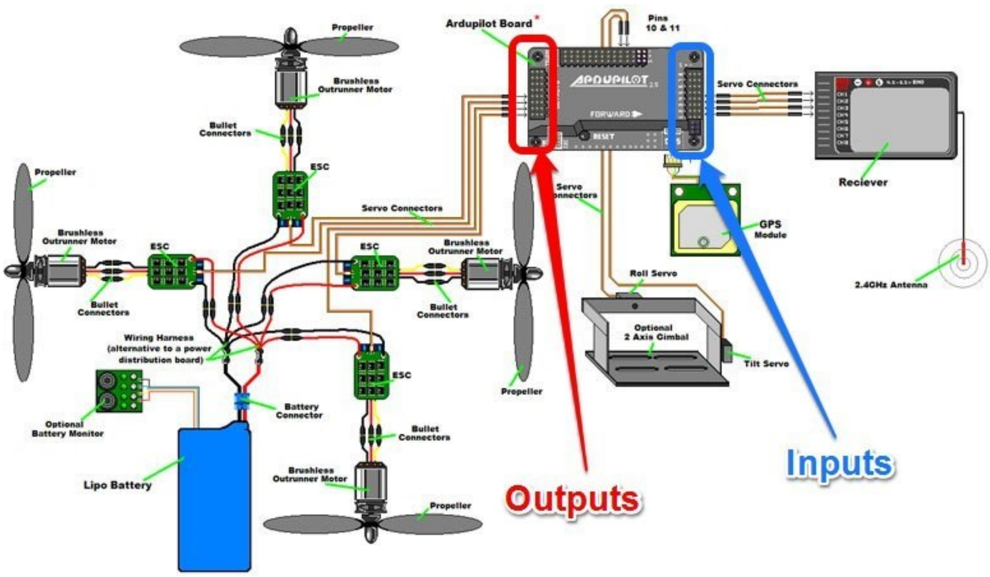


Figure 2.19: Example of sensors and actuators interaction on a quadrocopter layout [7].

2.5.3 Power Supply and Platforms

Small size UAV mostly use lithium-polymer batteries while the larger UAV rely on conventional airplane engines. The power supply for a specific UAV depends on its energy demands. Nowadays

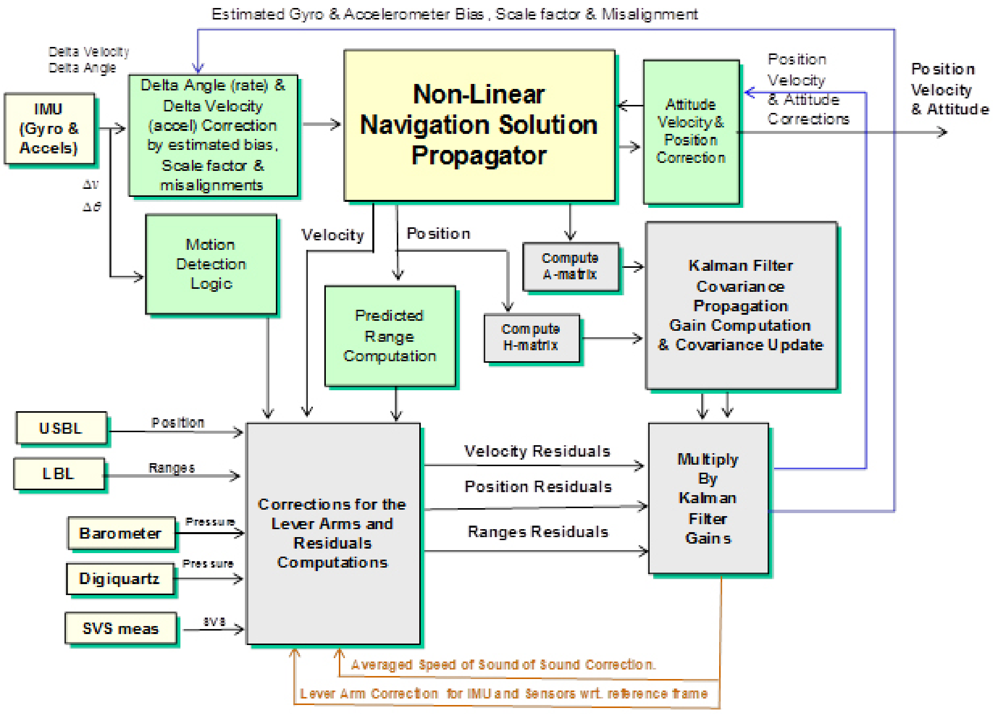


Figure 2.20: Example of main sensors main actuators interaction architecture. The green block are the actuators and the yellow block are the sensors. Other blocks are implemented controllers. From *Zupt*.

other technologies are being applied like solar power, or RF chargers, however power supplies still being one of the modern UAV bottlenecks. Also dependent on the morphology, platforms may differ but usually it stills the standard structure for UAV to park and recharge. This component is one of the most important because all other surrounding systems rely on it to be fully operational. In [45] and [6] a detailed conceptualization of Power supplies and their platforms is presented. Its important to refer once again that nowadays the power supply is the biggest bottleneck on drones autonomy once they can not stay long periods in the air.

On civil drones, the power system is basically a Li-Po Battery. In Fig. 2.19 the blue device is the battery device used to power supply the on-board actuators and sensors. In military drones fuel is the main use source of energy.

2.5.4 Controller

The central tenet over the UAS is that the operator is removed from the cockpit. Therefore, the control of the aerial vehicle must take place by other means. There are three forms of control that an operator my exert over the aircraft: Ground-Control or remote piloting, Semi-autonomous and autonomous. Ground-controlled UA also called RPV, require constant input from the operator. In 1980's and early 1990's system began employing both remote control techniques and programmable guidance systems. Thus the trend in UA circles has been towards more autonomous systems [5].

Semi-autonomous is the most common type of UA control. Semi-autonomous flight can be defined as requiring "ground inputs" during critical portions of the flight such as take-off, landing, weapons employment, and some evasive maneuvers". The operator must assume full control of the aircraft during pre-flight, take-off, landing, and when operating near base. However once airborne, an autopilot function can be engaged and the aircraft will follow a set of preprogrammed waypoints. The operator is responsible for the UA throughout the operation and can assume control at any time [20] [25] [5]. Fully autonomous UA capability lies at the other end of the spectrum. In theory, autonomous flight requires no human input in order to carry out an objective. An autonomous UA is able to monitor itself, assess its health, status and configuration; command and control assets on board of the vehicle within its programmed limitations [20].

In order to reach a high level of autonomy we need to understand first what are the UA's mission requirements and characteristics. Some missions can ask for dangerous difficult maneuvers or to fly under uncertain environment conditions, which make harder the drone to fly safely. Thus the trend is for fully autonomous UA, operator's surveillance over the system will be necessary for many years to come, once that even with AI developments, human decision capacity stills better at this moment [12]. In section 2.7 a more detailed autonomy level of UAS are exposed.

The three methods and control of UA referred before are achieved through different guidance, navigation, and control (GNC) systems. The less autonomous systems usually are guided and controlled through radio control. The semi-autonomous use essentially video based system and an autopilot. The fully autonomous UA use autopilot to fly autonomously and often incorporate algorithms for path planning and prediction, collision avoidance among many other cutting-edge algorithms that incorporate AI and neural networks, in order to learn from the experience [6]. In Fig. 2.22 is shown the classification of GNC system developed for drones based on their controllers autonomy, and control techniques [1] [6].

Beside these methods, almost any automatized UAV needs to employ some kind of control architecture. In open-loop architecture, the control is made without the feedback from sensor data. In closed-loop architectures sensor feedback is used to adjust the behavior of the UAV. It is also frequent to implement PID controllers in these kind of architectures. UAV can also employ hybrid architectures, where some inputs are controlled in open-loop and others on closed-loop. These architectures can be applied not only for UAV general control but also into more specific variables like navigation control, flight control, attitude control, velocity control, among others [6]. In Fig. 2.21 a control scheme is shown.

2.5.5 Ground Station

The ground control station (GCS) is composed by the UAV ground station, the off-board control computers, the UAV operator, and the on-board systems operator hardware, such as receivers, data links, and tracking stations/antennas. The communication between the UAV and the GCS is assured via a dedicated data link to the autopilot via the dedicated ground station (GS). It can have

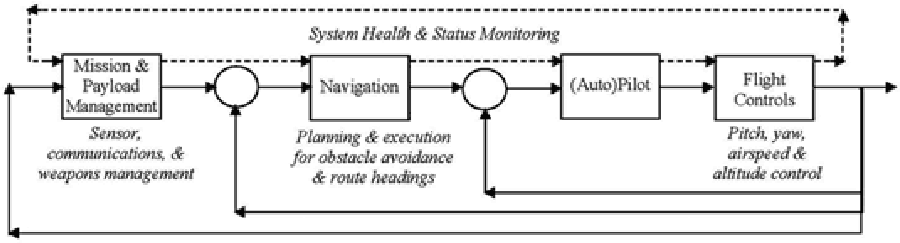


Figure 2.21: Hierarchical Control Loops example (for a single UAV) [8].

two different configurations, depending on the GCS – fixed or portable. The GS is connected to the operator's computer and the operator interface. The GS is also connected to a pilot console, in order for the pilot to fly the UAV, when in manual mode [5]. The distance between the GCS and the UAV depends on the communication range. Usually for small drones GCS is close and in LOS. For higher size and endurance UAV fixed GS are used because bigger and heavier equipment is needed to communicate into further distances so UAV can fly in BLOS mode.

One of the most important tasks realized by the GCS on the context of an UAS is to work as a network hub. Allowing and mediating data between GCS and the UAV on the sky.

GCS components and their specification are closely related with UAV requirements, autonomy level, network specification, payloads and mission.

2.5.6 Communication

Communication on UAS is one of the most important parts of the system, reason why term UAS started to being used instead of UAV. The communication architecture, the communicative technologies and the data link are crucial to obtain high levels of autonomy and reliability [46].

The principal issues of communication technologies are flexibility, adaptability, security, and cognitive controllability of the bandwidth, frequency, and information/data flows [20]. An UAS data link typically consists of a RF transmitter and a receiver, an antenna,and modems to link these parts with the sensor systems. In UAS, data links serve three important functions:

1. Uplinks from the ground station and/or a satellite to send control data to the UAV;

2. Downlinks from the UAV to send data from the on-board sensors and telemetry system to the ground station;

3. As a mean for allowing measurement of the azimuth and range from the ground station and satellite to the UAV to maintain good communications between them.

Efforts to standardize data links have resulted in the use of the common data link (CDL) protocol, typically a full duplex, wide-band data link that when used by UAS usually jam resistant and secure networks. These links connect the ground station with the UAV via direct, point-to-point links or use satellite communications (SATCOM) [47].

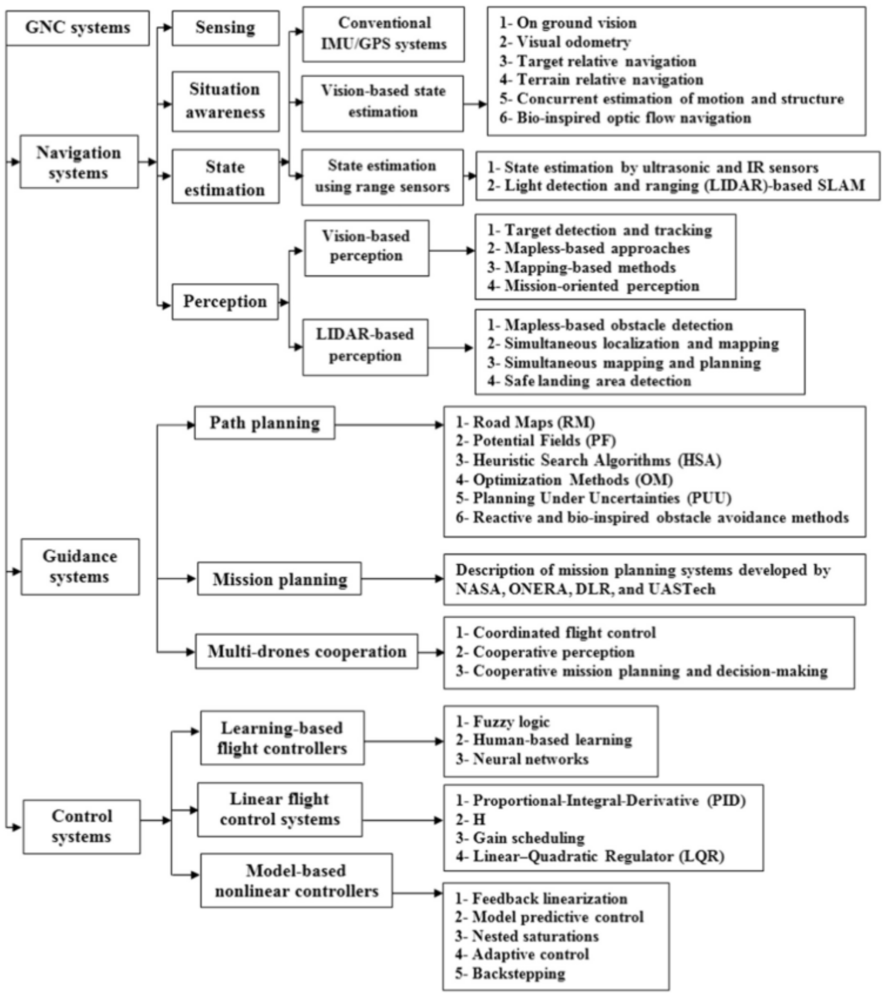


Figure 2.22: Classification of GNC systems developed for UAS based on [1].

The way we communicate its determinant for mission success. Through communication channel operator can remotely operate the UAV, read its flight informations, can read data from onboard components and its payloads. Furthermore he can also plan the mission and send the commands. On other kind of communication architectures the UAV can communicate with other homogeneous or heterogeneous drones and share information with each other [1]. This last kind of communication network is the trend of communication once drones get their applications amplified by the use of multi-agent systems and because would be impracticable at a certain level for a network to manage the constant calls to the server (ground station) if the amount of UAV is considerable big [48] [49]. Also, in this communication configuration big infrastructures are not needed.

The C3 system model is a standard model in UAS operations that stands for communication, command and control (C3). In Fig. 2.23 a C3 system breakdown is shown. Under each category of data links, UAS technical issues can be divided into two categories: Command & Control (C2) and Air Traffic Control (ATC) [9].

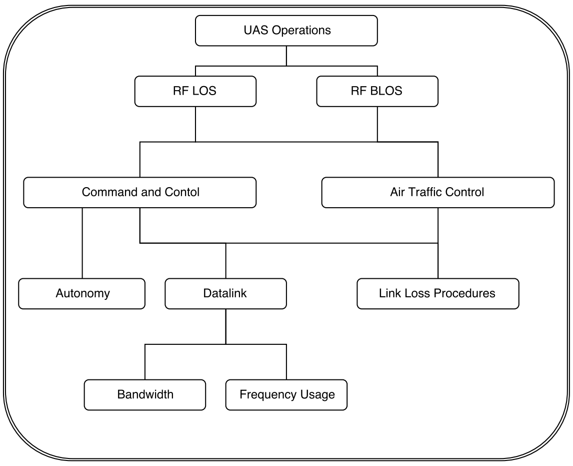


Figure 2.23: C3 system infrastructure from [9].

Inside those UAS data links they can be divided into two further categories: RF LOS and RF BLOS. An UAS data link consists of a radio frequency transmitter and receiver, an antenna and a modem to link all these components with the UAV [38].

BLOS data link contains LOS technologies along with satellite-based communications (SATCOM). SATCOM is used for C2 of the UAV and for communications with Air Traffic Control (C3). As the distance between the GCS and the UAV increases, the signal-to-noise ratios of the transmission link will fall. The signal loss may manifest as a lag in the UAV command uplinks and recovery of data downlinks [21] [20]. In Table 2.2 different possible frequency bands are shown.

Table 2.2: List of frequency bands used in SATCOM [21].

Band	**Frequency**
HF	3-30 MHz
VHF	30-300 MHz
UHF	300-1000 MHz
L	1-2GHz (General) 950-1450 MHz (IEEE)
S	2-4 GHz
C	4-8 GHz
X	8-12 GHz
Ku	12-18 GHz
K	18-26.5 GHz
Ka	26.5-40 GHz

As previously referred, not all communication architectures are directly UAV to GCS. Some use network centric technologies like the one's in [20]. One of those is a MANET, which is a flexible wireless network applicable to an homogeneous UAS team that does not require any infrastructure to operate. This kind of infrastructure-less network oriented to collaboration is known as Mobile Ad-hoc networks (MANET). This kind of networks are starting to be extensively used in military and civil applications once on this kind of networks, every node acts as a communicating repeater, forwarding information to the destination node. This allows more traditional data link communication systems to be less overloaded with "cheap" information [20]. In Fig. 2.24 examples of different communication data links are shown. C3 model is also visible, and main components and its interactions it is also perceptible for different situations.

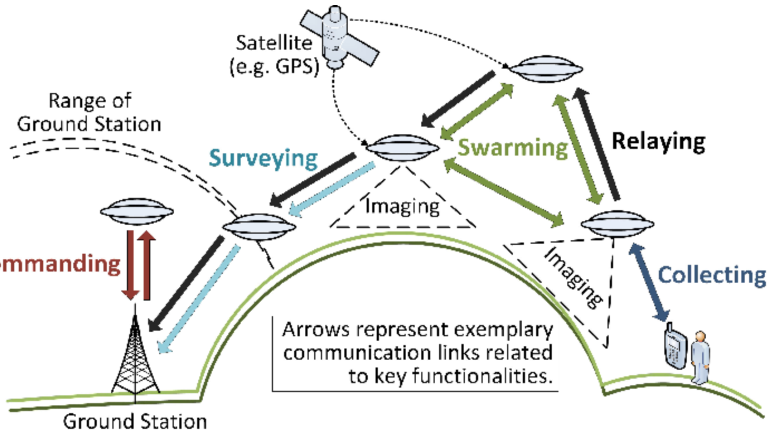


Figure 2.24: Example of an UAV Communication System using different data links [5].

For further information about this topic [20], [21], [6], [49], [46] and [47] are recommended.

2.5.7 Software

UAV software is composed by the operating system (OS), the Application Programming Interface (API), the communication protocols and autopilot. These components are very important not only from the point of view of execution, but also in ease of system development. Due to time critical constraints imposed by the nature of flight control, high reliability and real-time execution are mandatory in UAVs software.

The software is the interface where processors gather and manage the information. Hardware without Software are like Humans without brains. They have the sensors collecting data however they can not be programmed to react to their variation. Nowadays, we have several Softwares and platforms for mission planing and UAV tracking which allows better interaction with operator and more accessibility to this technology. Furthermore, those units perform all tasks that UAV's Hardware need to operate.[5] From the ground station, those units have the same operation properties but are usually more operator's oriented. The softwares are more customized, and the process power is higher allowing the control of more than one UAV at the same time. From Fig. 2.25 we see that software modules can be mainly divided into 3 layers. The firmware, middleware and the operating system. Their requirements are time-critical for the two first layers and computer intensive for the last one. Example of operations are machine code to processor, Flight control and collision avoidance respectively [1]. [20].

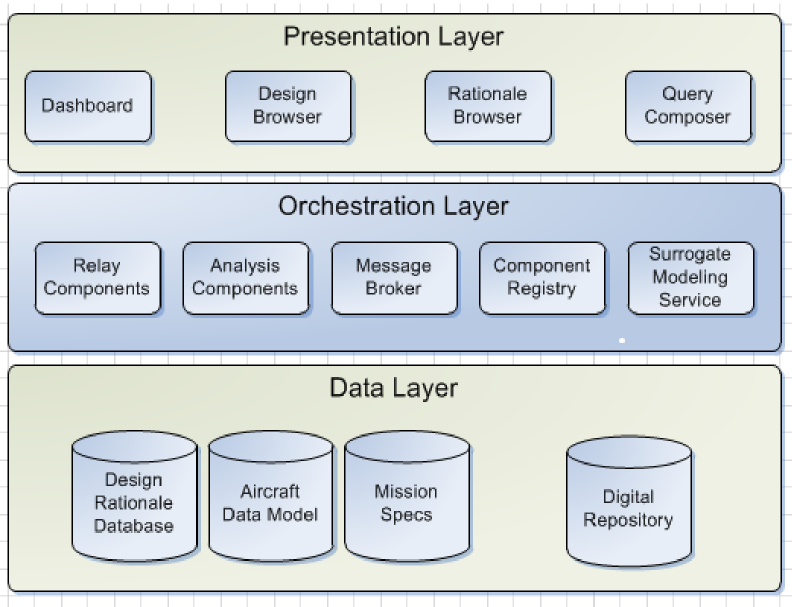


Figure 2.25: UAS software layer and function examples [10].

Table 2.3: Example of operation an open source API for each software layer in UAS.

Layer	Requirement	Operations	Example
Firmware	Time-critical	From machine code to processor execution, memory access. . .	ArduCopter-v1.px4
Middleware	Time-critical	Flight control, navigation, radio management...	Cleanflight, ArduPilot
Operating system	Computer-intensive	Optic flow, obstacle avoidance, SLAM, decision-making...	ROS, Nuttx, Linux distributions, Microsoft IOT

As shown in Table. 2.3 nowadays a lot of open-source OS and API exist to research issues or even for recreational use. Further on this thesis some examples of UAS softwares are shown. Some examples are autopilots, mission planning simulators or even vehicle control simulators [50].

2.6 Flight Dynamics

Flight Dynamics is the science of air vehicle orientation and control in three dimension, in this case, all physics and dynamic models associated with our UAV. The dynamic of an UAV varies accordingly to its specifications and requirements. Changes in its body's shape and size imply different movements laws and equations which requires different controllers as well. Is important to study UAV models so we can design high performance controllers and boost UAS overall performance.

In terms of a basic dynamic, UAV three critical states are the angles of rotation. These angles of rotations are designated as Roll, Pitch and Yaw. Each one of these angles represent the rotation of the axis over the center of mass. This maneuvers can happen independently or mixed with the others. In Fig. 2.26 is showed the rotation of an UAV over the three main axis. We can see that the rotation of the aircraft over longitudinal axis is called Roll, the rotation over the lateral axis is called Pitch and the rotation over the vertical axis is called Yaw.

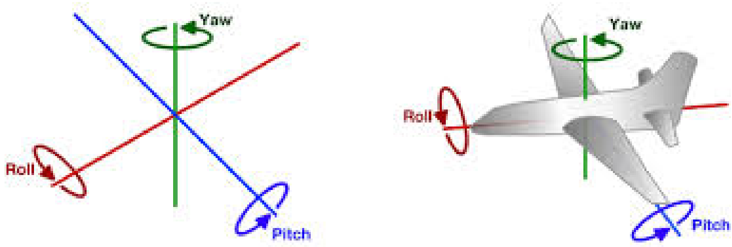


Figure 2.26: Example of the angles of rotation in an aircraft [11].

In order to obtain the orientation and control the UAV a coordinate as to be established. In [11] two frames are considered: $E = \{E_x; E_y; E_z\}$ earth fixed frame and $B = \{B_x; B_y; B_z\}$ as the UAV's body fixed frame. This way we know the position and orientation of B relative to E. In [12] the same principle is applied with different variables designation as shown in Fig. 2.28 but for now lets focus on the considerations made in [11] and shown in Fig. 2.27.

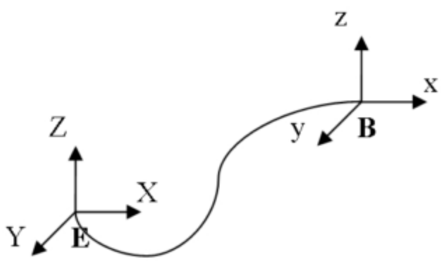


Figure 2.27: Coordinate System in [11].

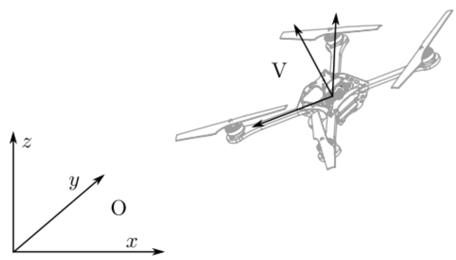


Figure 2.28: Coordinate system in [12].

The rotation of UAV body in space is then parameterized using several methods (Euler angles, Tait-Bryan angles, among others). In Figure 2.29 we can see the Tait-Bryan rotation and in Eq. 2.1 we see the mathematical representation of the three successive rotations about different possible axes accordingly to Tait-Bryan angles, also called "Cardano angles", which are extensively used in aerospace engineering where they are called "Euler angles" (the true "Euler angles" are designated as "classic Euler angles") [11] [43].

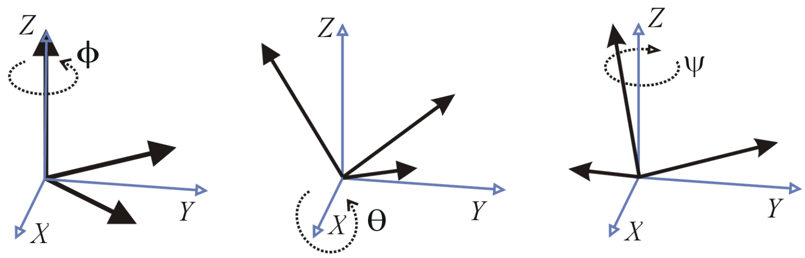


Figure 2.29: Successive rotations in roll, pitch and yaw according to Tait-Bryan convention.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} R_z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Using the Tait-Bryan angles to describe the orientation of the UAV, consists in using three successive rotation: Roll, Pitch and Yaw. These three rotation are of ϕ around $\vec{x}$, θ around $\vec{y}$ and ψ around $\vec{z}$, respectively. Originating angles for roll of $-\pi < \phi < \pi$, for pitch of $-\pi/2 < \theta < \pi/2$ and for yaw of $-\pi < \psi < \pi$.

The complete rotation matrix is represented in Eq. 2.2 where we get all the separate rotations into one, obtaining a complete rotation matrix, also denominated *Direct Cosine Matrix* [51]. This rotation matrix represent the different three dimension state of an Unmanned Vehicle relatively to a coordinate system.

$$\begin{aligned} R(\phi,\theta,\psi) &= R(x,\phi)\cdot R(y,\theta)\cdot R(z,\psi) \\ R(\phi,\theta,\psi) &= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\sin\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi - \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \end{aligned} \quad (2.2)$$

The time variation of Tait-Bryan angles $(\dot{\phi},\dot{\theta},\dot{\psi})$ is a discontinuous function. Thus, it is different from body angular rates (p,q,r) which are physically measured with gyroscopes.

In general, an IMU is used to measure the body rotations and directly calculate the Tait-Bryan angles. This way we can get [52]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_r \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.3)$$

Where p,q and r are the IMU roll rate, IMU pitch rate and IMU yaw rate respectively.

$$R_r = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{2.4}$$

Despite the Tait-Bryan angles representation suffer from a singularity know as the "gimbal lock" UAV normal flight mode does not get affected.

Note that these formulas are general for any kind of vehicle, and they correspond to the orientation method for the UA. Thus, in order to obtain the specific UA control laws, despite they are based on last formulas they also include other factors has the propellers positioning, the UA forces equilibrium, aerodynamic, among others. Only having calculated all those points we can get to design a flight controller. The equation of motion of the different UAV are basically the translational equations alongside the rotational equations. The most common kind of UAV which are the Fixed-Wing UAV and the rotary-wing UAV have different dynamic models. However, the basic principle is the same [11] [43].

For instance, the quadrocopter with the coordinated system presented in Fig. 2.28 is described by six degrees of freedom: the translational position (x,y,z) is measured in the inertial coordinate system O. The vehicle attitude V is defined by the rotation matrix ${}^O_V R$ like the rotation matrix presented in Eq. 2.4. The rotation matrix is defined such that, when multiplying a vector v in the coordinate V system with it, the same vector, described in the inertial coordinated system O, is obtained:

$$^{O}v = {}^O_V R \cdot {}^{O}v \tag{2.5}$$

Eq. 2.3 and 2.5 are the same, expect the coordinate system is different. The control inputs of the quadrocopter are presented in Fig. 2.30 where w_x, w_y and w_z represent the desired rotational rates of the quadrocopter body axes, and a the mass-normalized collective thrust.

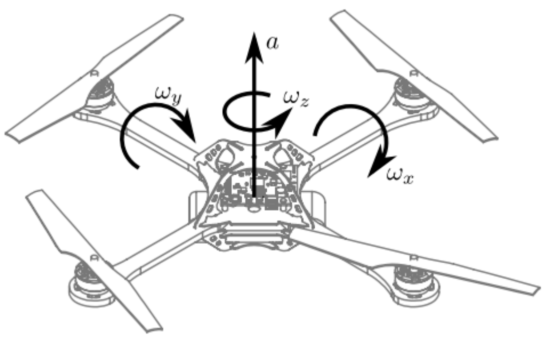


Figure 2.30: The control inputs of the quadrocopter. The value of these inputs are controlled by on board sensors and gyroscope feedback [12].

Result is that the translational acceleration of the vehicle is dictated by the attitude of the vehicle and the total thrust produced by the four propellers. The translational acceleration in the inertial frame is:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = {}^{O}_{V}R \cdot \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{2.6}$$

The change of vehicle attitude is related to the rotational control inputs through [12].

$${}^{O}_{V}\dot{R} = {}^{O}_{V}R \cdot \begin{bmatrix} 0 & -w_x & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{2.7}$$

Like seen in this quadrocopter example from [12], the Eq. 2.6 and 2.7 are the equations of motion, which are purely constructed over the UAV model and the orientation control. Another example of equations of motion this time for a fixed-wing UAV is seen in [11]. Further and different equations of motion and controls are shown in the literature [53], [54], [55], [51], [52], [43], [56] and [57].

2.7 Autonomy

Autonomy level of a system is directly related with the task to be performed. In [20], UAS may be automated, autonomous, semi-autonomous and piloted remotely or a mixture of those capabilities [50]. A minimal automatic control of the flight involves attitude sensors and on board processors. Due to the high non linearities of the flight dynamics, a set of control techniques (PID, fuzzy logic, neural networks, among others) are used in autopilot systems to guarantee the desirable navigation. In [50] the analyzed literature highlights that there is not a unique definition of the concept of the autonomy applied to unmanned vehicles. Nevertheless, all the definition found share a common philosophy which can be summarized in the following statement: *"Autonomy is the ability of an agent to carry out a mission in an independent fashion without requiring human intervention."*

Often, the concepts autonomy and automation are misused to describe the ability of a system to operate without human supervision. An automated system cannot choose freely because all its decisions are previously programmed. On the other hand, an autonomous system is able to make its own decisions, having a certain level of "freedom". Illustrative example of an automated system is an autopilot system that keeps the UA within pre-established path. On the other way, a collision avoidance system, which is able to resolve air conflict risk by itself is considered an autonomous system. Despite the difference on the use of those two concepts, they are related once the level autonomy is directly related to the level of automation of certain system. Similarly, autonomy and intelligence are distinct concepts as well. The latter, is defined as the ability of an agent to learn new concepts and to apply the acquired knowledge to the execution of a new task. This ability has nothing to do autonomous level. Nature is the best example to corroborate the last statement, once in nature there are autonomous systems with no intelligence, like the microorganisms. Artificial intelligence of UAS its widely studied in some literatures like [14].

Based on human cognitive process, in the current state-of-art there are four steps of the information processing cycle that are suitable to incorporate a certain form of autonomy or automation. These steps are known as OODA that stands for Observe, Orient, Decide and Act cycle. Any kind of autonomy applied to these steps result into system functionality that is eventually automated. In Fig. 2.31 the human cognitive cycle model is presented.

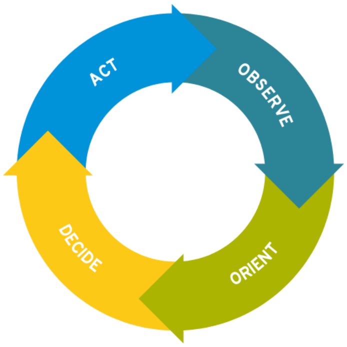


Figure 2.31: Human cognitive Model.

Several factors have to be considered in order to select the optimum level of autonomy for a UAS team in any context. The requirements of the mission play a major role in such a selection but there are other relevant aspects such as environment, efficiency or cost.

There are two types of UAS operational control systems: executive and supervisory depending on the operator's involvement. Most UAS endowed with a minimum degree of autonomy are controlled by at least one operator who plays as supervisory [50]. Indiscriminately high autonomy levels do not yield satisfactory results in applications or scenarios that are subject to critical time constraints. On the other hand, insufficient levels of autonomy generate an excessive workload that absorbs the operating capacity from the human operator and reduces the efficiency of the mission. Therefore, an equilibrium must be sought between human action and low level tasks automation to achieve an optimum performance [58], [59]. To choose the level of autonomy on our UAS system, we need to analyze first the mission demands and see some aspects like team-size, autonomy levels, automation reliability, task coordination and allocation, scalability and collateral factors [50].

Knowing these issues, one way to determine the autonomy level of an UAS is by using a methodology known as metrics that constitutes a way to assess both quantitatively and qualitatively the amount of autonomy that a vehicle or a team of vehicles possesses [14].

Autonomy Levels of Unmanned Systems (ALFUS) is a three axis metric framework to determine autonomy level and operator dependence or independence in the mission to be performed [13]. This framework, presented in 2004 SPIE Defense and Security Symposium, is composed by mission complexity, environmental conditions and human interface. As shown in Fig. 2.32 the different challenges to the autonomy level are disposed in a three axis framework were each one is classified with a metric. Each metric it is weight associated since some can contribute more or less to the overall autonomy of the system than others. For that, the weight of each metric of an Unmanned System (UMS) is computed to determine the degree of autonomy of that system.

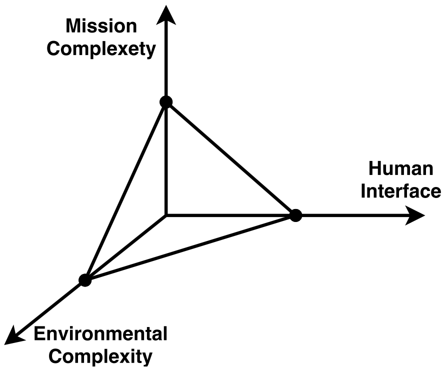


Figure 2.32: ALFUS framework.

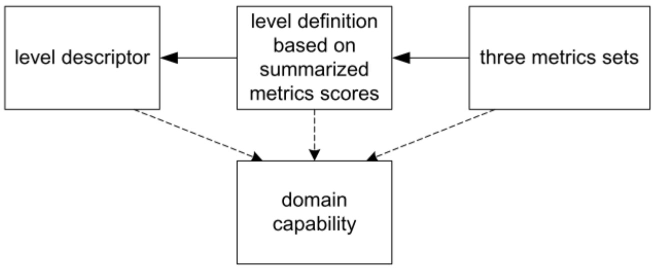


Figure 2.33: Alfus Model Summary [13].

This model derives from the Fig. 2.33 model overview. Starting from the right hand side, we have a summarization of the metric values for the particular autonomy levels. This is done at a generic level of abstraction. For example, at the highest level of autonomy, the environmental difficulty should be extremely high and the HRI should be approaching minimum. We, then, derive definitions for the levels from the metric summary, as the top-middle box in the figure indicates. Third, we create descriptors for the levels. The purpose is to facilitate human communication. The generic summary model can be applied to particular domains to identify specific mission and task capabilities as well as particular autonomy level scales, as the bottom box shows [13].

In Fig. 2.34 the resulting work of applying the Figs. 2.32 and 2.33 model is shown. Note that, while color bands are used to delineate a continuum of different autonomy levels, there is no implication of distinctive switching of autonomy capabilities at the color boundaries. The general trend is that, the more a robot is able to see, learn, think, plan, and act independently or collaboratively, to achieve assigned complex goals in difficult environments, the higher the level of autonomy the robot should have. Whether the robot performs the mission with or without *a priori* knowledge that does not affect its level of autonomy [13]. Also note that in this model, the lesser is the human-robot interaction (HRI), the higher is the collaboration between agents, mission and environment, as expected, the higher is the autonomy level of our system. The features that make the UAS achieve this levels are technologies such as automation, autonomy, intelligent flight planning, fault monitoring, sense-and-avoid, guidance and flight control, navigation, payload management, recovery, operative system, among many others [20] [6] [50].

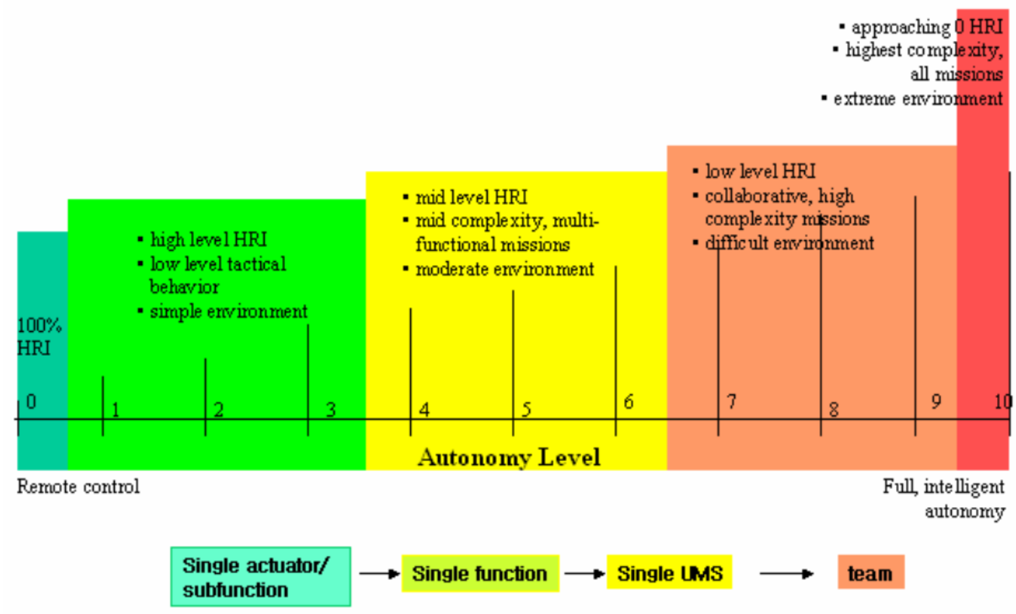


Figure 2.34: ALFUS Summary Model Overall Concept [13].

Nowadays trend and military and market necessities from unmanned system involves a high degree of human interaction. The objective is to continue develop and pursue new technologies and policies that introduce a higher degree of autonomy to reduce the manpower burden and reliance of full-time high-speed communication links while also reducing decision loop cycle time. The introduction of increased unmanned systems autonomy must be mindful of affordability, operational utilities, technological developments, policy, public opinion, and their associated constraints. Though still a long road until full autonomous vehicle, new self-trained algorithms, software hierarchy, neurological and genetic algorithms are the next big step in that direction.

In Fig. 2.35 (chart presented in [14]) shows a roadmap of autonomous level of UAS is displayed. As can be seen, the autonomous level of UAS shows an exponential growth which supports that technologies development, specially on this area, is of increasingly importance thanks to the application that can be achieved through the integration of not only artificial intelligence but also ad-hoc networks in the current UAS. For more knowledge about this issue consult [6], [13], [14], [20], [50], [58] and [60].

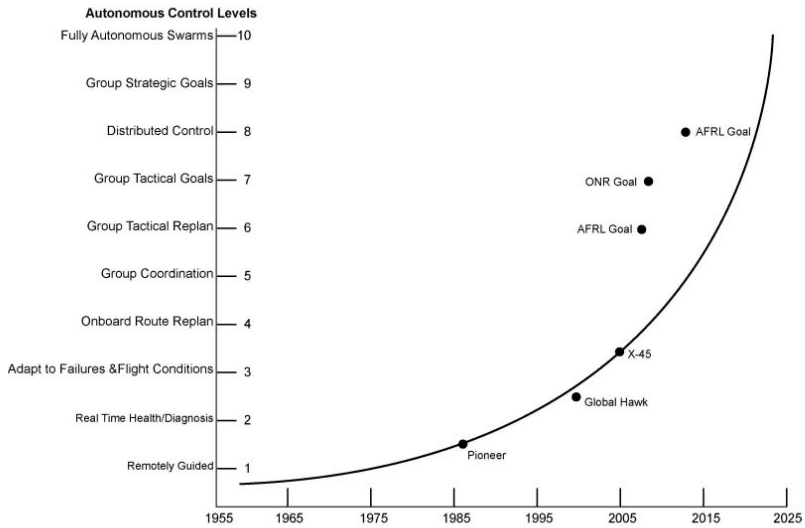


Figure 2.35: Autonomous Control Level Roadmap [14].

2.8 Applications

The application of drones cover a wide range of civil and military scenarios. Drones can perform both outdoor and indoor missions. As shown in [6] drones application can be categorized in different ways. It can be based on the type of missions, type of flight zones and type of environments. Like shown in Fig. 2.36, drones have a wide variety of applications in daily life. Its estimated on [61] and [62] that drones can have more than two-hundred application in near future. Classic operations include: environmental exploration and sampling, sensor deployment, terrain coverage, air and ocean monitoring and patrolling, search and rescue missions, mailing and delivery, among many others [63] [6]. In Table 2.4 we see many other examples of applications for drones or teams of drones.

2.9 Multi-robot Coordination

From [23] we get that in the field of mobile robotics, the study of multi-robot systems (MRSs) has grown significantly in size and importance in recent years. Having made great progress in the development of the basic problems concerning single-robot control, many researchers shifted their focus to the study of multi-robot coordination.

Today's robots fall into one of three main categories: namely manipulators, mobile robots and humanoid robots.

2.9.1 Robotic Systems: Single-robot vs Multi-robot

A single-robot system contains only one individual robot that is able to model itself, the environment and their interaction [64]. The robot in a single-robot system is often designed to deal

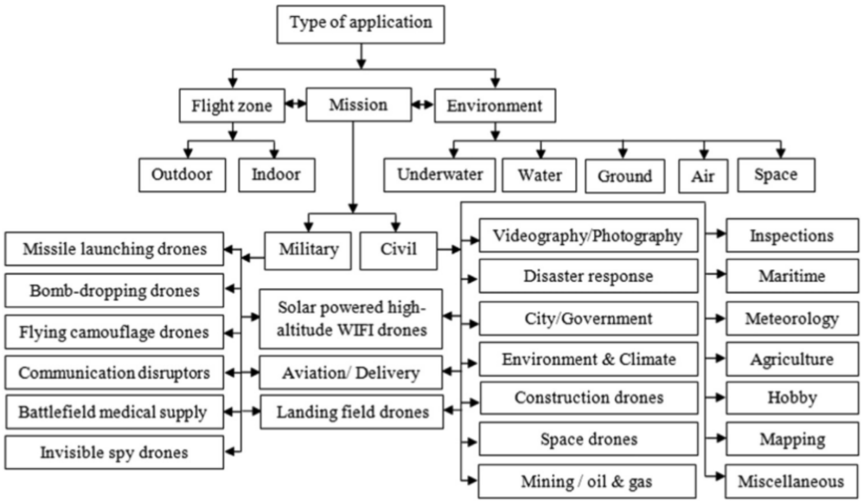


Figure 2.36: Examples of drones application categorized accordingly to [6].

with a task on its own account. Such robots are usually integrated with multiple sensors, which themselves need a complex mechanism and an advanced intelligent control system. Although a single-robot system give have a relatively strong performance, some tasks may be inherently too complex or even impossible for it to perform, such as spatially separate tasks. Hence, single-robot system is spatially limited. This means that for our work of surveillance, multi-robot perform better than just one.

On other hand, MRS contains more than just one individual robot. Those groups might be homogeneous or heterogeneous. This brings several potential advantages when compared to a single-robot system such as [23]:

- MRS has a better spatial distribution.
- A MRS can achieve better overall system performance. The performance metrics could be the total time required to complete a task [65] or the energy consumption of the robots [66].
- A MRS introduces robustness that can benefit from data fusion and information sharing among the robots, and fault-tolerance that can benefit from information redundancy. For example, multiple robots can localize themselves more efficiently if they exchange information about their position whenever they sense each other [67] [68] [69].
- A MRS can have a lower cost. Using a number of simple robots can be simpler (to program), cheaper (to build) than using a single powerful robot (that is complex and expensive) to accomplish a task.
- A MRS can exhibit better system reliability, flexibility, scalability [70] and versatility. Robots with diverse abilities can be combined together to deal with complex task, and one or several robots may fail without affecting the task completion.

Table 2.4: Examples of drones application accordingly to [22].

Remote Sensing	**Surveillance**	**Comm Relay**
-Pipeline Spotting -Powerline Monitoring -Volcanic Sampling -Mapping -Meteorology -Geology -Agriculture	-Law Enforcement -Traffic Monitoring -Air/Ocean Patrol -Border Patrol	-Internet -Cellular Phone
	Transport	**Entertainment**
Search and Rescue	-Cargo Transport	-Cinematography -Advertising
Disaster Response	**Delivery**	**Broadcast**
-Chemical Response -Flood Monitoring -Wildfire Management	-Firefighting -Crop Dusting -Package Delivery	-Television/Radio

Homogeneous robot systems mean all robots capabilities are identical. Heterogeneous robot systems mean that robots capabilities are different. The second system trends to be generally more complex because as the robot have different capabilities it becomes more difficult the task planning and the compatibility between robots.

2.9.2 Multi-robot: Cooperative vs Competitive

Multi-robot environments can be cooperative or competitive. Cooperation refers to a situation whereby multiple robots need to interact together to complete a task or increase the system efficiency. Alternatively, cooperation is the interaction between the robots, which work towards a common interest or reward. The cooperative robots have a joint goal, which gives rise to various sub-goals.

Competition refers to a situation whereby robots compete against each other to best fulfill their own self-interest. Alternatively, robots with conflicting utility functions are in competition with each other [71]. The competitive behavior is the opposite of cooperative behavior.

Cooperation examples are multi-robot exploration, localization and transportation. Competitive examples are for example robot soccer leagues or chess games. In order to boost multi-robots systems advantages, we need to find a sort of consensus, this is a decision making mechanisms so we can optimize multi-robot systems efficiency and yield.

Robots might be selfish from the sociological point of view, because a single robot tends to make decisions motivated by self-preservation. For instance, consider two robots moving in opposite directions and wanting to cross a narrow passage, but where only one may cross at a time. If the two robots move simultaneously, a congestion or collision will occur. The cooperation can overcome group think and individual cognitive bias, and this requires some form of coordination.

Such coordination can be achieved by communication, which is often used as a rational behavior in multi-robot environments.

If multiple requests targeting the same resource arrive simultaneously, resource conflict will occur. In MMRS, resource conflict arises when multiple robots need to share space, manipulable object or communication media. To deal with this problem, we need to coordinate the robots. This coordination can be reflected in task and motion planning.

Another conflict is communication: as previously mentioned, communication is important for a MRS because it can help robots to be cooperative by learning information that is observed or inferred by others. However, the communication media cannot always be shared, therefore it is necessary for the robots to obtain exclusive access to them. The problem of communication media sharing is often associated with bandwidth limitation. To solve this problem [72] presented a method to evaluate control and communication strategies for a group of wireless-networked robots. In this method, they used a network simulator [73] to handle the bandwidth limitation problem, which took into account protocol characteristics and propagation conditions. [74] described a distributed software control architecture designed for the control of multiple robots over a low bandwidth communications channel. In this architecture, robots share communication bandwidth by using the time slot (time-division multiplexing).

Other conflict types of conflicts are box-pushing problem and space sharing problem. Like the other conflicts type researchers try to develop new architectures and optimize tasks planning in order to avoid this conflicts and find the perfect balance.

2.9.3 Static vs Dynamic

Multi-robot coordination is the core task of MRSs. The overall system performance can be directly affected by the quality of coordination and control. Coordination can be static or dynamic. Static coordination generally refers to the adoption of a convention prior to engaging in the task. Dynamic coordination occurs during the execution of a task, and is generally based on the analysis and synthesis of information. The information can be obtained through the means of communication.

Dynamic coordination can also be divided into two categories: explicit coordination and implicit coordination. Explicit coordination is usually associated with explicit communication, which is produced by a robot's active behavior. Implicit coordination is usually associated with implicit communication, which requires the robot to perceive, model and reason others' behavior.

The static method can handle complex tasks, but its real time controlling might be poor. The dynamic method can well meet the capability of real-time, but it has difficulty in dealing with more complex tasks. As a result, the optimal solution lies between the balance of both methods.

2.9.4 Explicit vs Implicit

Communication, as a means of coordination, often emerges as a rational behavior in multi-robot environments. In fact, the communication is a mode of interaction between robots. By this interaction, on the one hand, robots can share position information, the state of the environment and sensor data with others in the system; on the other hand, an individual robot can get information as to the intentions, goals and actions of other robots. At [75] classified the communication structure into three types according to the mode of interaction, which includes: interaction via the environment, interaction via sensing and interaction via explicit communications. In [76] they distinguished two different types of communication depending on the way in which the robots exchange information, which includes direct and indirect communication.

Explicit communication refers to the means for the direct exchange of information between the robots, which can be made in the form of single cast or broadcast intentional messages. Implicit communication refers to the way in which the robot gets information about other robots in the system through the environment. The use of explicit communication can ensure the accuracy of the exchange of information between robots. However, the communication load of a system will increase as the number of robots increases. This may cause a decrease in system performance or else lead to an overall system failure in extreme cases. In using implicit communication, and although the information obtained by a robot is not completely reliable, the stability, reliability and fault tolerance of the whole MRS are better than in using an explicit pattern. Therefore, applying both explicit and implicit methods in practice can make the two methods complement each other.

2.9.5 Task Planning and Motion Planning

Multi-robot planning is usually divided into two aspects: task planning and motion planning. Task planning is primarily designed to solve the problem of which robot should execute which task. Motion planning is primarily designed to generate the path of each robot.

In robotics, the motion planning problem involves producing a continuous robot motion from one configuration to another in a configuration space while avoiding collision with obstacles. Motion planning is eminently necessary for mobile robots since, by definition, a robot accomplishes tasks by moving in the real world [77].

Multi-robot motion planning (MRMP) should consider not only any obstacles (whether static or dynamic) in the environment, but also any possible interference between robots. This is because, when robots in a team are used to perform independent tasks in a shared workspace, each one will become a mobile obstacle for the others. Therefore, the motion planning of each individual robot in the team should take into account the movement of others. Even for a simple case in two dimensions, the problem is NP-hard and not solvable in polynomial time [78]. Most of the existing approaches to MRMP are expanded from the results of a single-robot system. Three major families of approaches are the cell decomposition, potential field and roadmap approaches. They all reduce the continuous motion planning problem to a discrete graph search problem by identifying some canonical states and paths within the free space.

The cell decomposition approach decomposes the free space into a finite number of contiguous regions, called cells. This approach is often applied for the multi-robot area coverage problem.

The roadmap approach reduces the robot's free space to a set of one-dimensional curves connecting a set of nodes, called a roadmap. A typical roadmap approach is a Voronoi diagram, which specifies the set of all points equidistant from two or more closest obstacles. Following the Voronoi diagram may not give the shortest path, but the resulting paths tend to maximize clearance [79].

Another roadmap approach is the probabilistic roadmap [80], which has been widely used for robot arms in engineering and manufacturing. This method randomly generates a large number of collision-free configurations and achieves motion planning by connecting some of them. Several studies address multi-robot coordination based on probabilistic roadmap, but focus on manipulator arms [81] [82].

Another probabilistic method is the rapidly-exploring random tree [83], which takes the motion planning as a tree search problem. The tree is constructed incrementally in such a way that any random configuration in the free space is added by connecting it to the closest configuration already in the tree.

The existing motion planning methods have their own strengths and weaknesses. Using only one method may entail difficulties. Combining multiple methods together is more coincident with the MRS requirements of reliability, flexibility, scalability and versatility [23].

2.9.6 Centralized vs Decentralized

In MRS, the decision making guided by planning can be centralized or decentralized in accordance with the group architecture of the robots. The advantage of the centralized architecture is that the central control agent has a global view of the world, whereby the globally optimal plans can be produced. Decentralized architectures can be further divided into two categories: distributed architectures and hierarchical architectures. In contrast to a centralized architecture, a decentralized architecture can better respond to unknown or changing environments, and usually has better reliability, flexibility, adaptability and robustness. Nevertheless, the solutions they reach are often suboptimal [23].

2.10 LSTS

From LSTS official website [15]:

The Laboratório de Sistemas e Tecnologia Subaquática (LSTS) is an interdisciplinary research laboratory established in 1997 with researchers drawn from Electrical and Computer Engineering, Mechanical Engineering and from Computer Science.

The LSTS is specialized on the design, construction, and operation of unmanned underwater, surface and air vehicles and on the development of tools and technologies for the deployment of networked vehicle systems.

In the last 20 years researchers from the LSTS have successfully fielded unmanned air, ground, surface and underwater vehicles in the Atlantic and Pacific oceans, and also in the Mediterranean sea. These vehicle systems implement the LSTS control architecture with the help of the LSTS Neptus-IMC-Dune software toolchain. Neptus is a distributed command, control, communications and intelligence framework for operations with networked vehicles, systems, and human operators.

IMC is a communications protocol that defines a common control message set understood by all types of LSTS nodes. DUNE is the system for vehicle on-board software. It is used to write generic embedded software at the heart of the vehicle, e.g. code for control, navigation, or to access sensors and actuators. It provides an operating-system and architecture independent C++ programming environment for writing efficient real-time reactive tasks in modular fashion. The toolchain has support for Disruptive Tolerant Network (DTN) protocols.

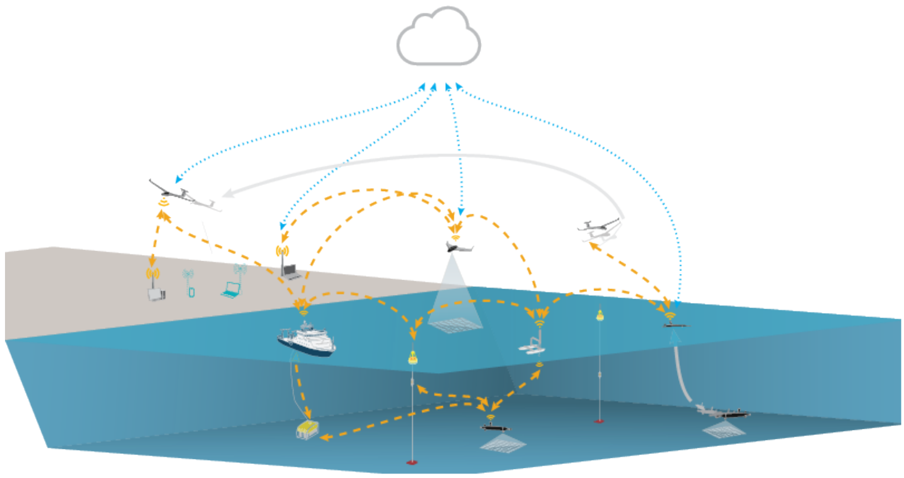


Figure 2.37: LSTS System and Toolchain Architecture and Interaction from [15].

2.11 Conclusion

On this chapter all the variety of topics related to UAS were exposed or referred. Starting from defining what an UAS is we exposed the history of UAS and their evolution. After, we shown the UAV and UAS components and they key characteristics and varieties. Therefore, flight dynamics was introduced and with it, UAS autonomy taxonomy and measuring. Finishing the technical exposure, UAS attractive applications were presented. As result of UAS applications we leaped into a brief discuss about the higher overall performance of UAS teams and analyze how different multi-robots coordination sets work and compared them with each other. For last, we presented the LSTS which is where this thesis was inspired on and where some SITL experiments were driven.

Chapter 3

Literature Review

In this chapter a literature review will be presented. Based on the assets presented on Chapter 2 and the multi-agent area coverage systems we want to implement all the relevant researches made on the topic will be briefly summarized.

We will start by defining surveillance, area coverage types and strategies. Next, we will talk about in area decomposition methods for sharing the work load and the area responsibility among all the agents in the system. Then, we will summarize the work developed in the different systems field, the non-Communicative area coverage and the communicative area coverage. As last we talk about related works and summarize the chapter.

3.1 Surveillance

Accordingly to [24] surveillance its by definition the monitoring of behavior, activities, or other changing information for the purpose of influencing, managing, directing, or protecting people. Surveillance activities are often used by governments for intelligence gathering, prevention of crime, the protection of a process, person, group or object. It is also used by business and civilians to gather intelligence, to patrol industrial facilities, plantation or agricultural fields. Applications, such as environment monitoring, search and rescue, precision agriculture and surveillance, may benefit from this usage of UAS with on board sensors for spatial coverage [84] [85] [86] [87].

Furthermore, to patrol it is also one surveillance technology and accordingly to [63], patrolling can be defined as the act of walking or traveling around an area, eventually at regular intervals, in order to protect or supervise it. In general sense, it may involve the task of visiting selected places in a given area in order to assess the state of the environment in relation to the presence of intruders or the occurrence of unforeseen events.

Despite there are other kinds of surveillance strategies. Classic area covering, monitoring and exploring are our priority. Most of the works referred on this chapter include their own point of view of what surveillance task is. However the main definition is exposed above.

3.2 Area Coverage and Area Partitioning

The purpose of area coverage in UAS is to sense a region of interest with equipped cameras or other type of sensors. The agent may take a video or pictures of an observed region to provide efficient overview to a base station [88].

Most of area coverage proposals using multi-drones team rely on cooperative operation architecture because they enhance the overall performance of the tasks [89] [88] [90]. If the agents cannot monitor all the terrain at all time, they must visit the target area periodically. This means that, accordingly to [91], we can have two different methods to surveil the target, a *coverage-oriented* patrol or *on-demand* patrol. The [91] work, involves visiting all target points as frequently as possible, while the latter involves monitoring target points to detect an intrusion by any detected event.

When the number of agents available is not enough to spread over the region, the agents have to perform area coverage non-communicatively. However when there are enough agents to entirely cover the region of interest, MANETs can be applied and communication is possible [88].

In area coverage without communication, the task for each agent is subdivided disjointly. In this kind of coverage each agent complete is task by sweeping and sensing its assigned region preferably efficient and quickly as possible. This is done by successive back and forth movements. Once sharp turns require big deceleration and accelerations they are the main responsible for energy and time consumption[88] [17] [92]. Despite the rotary-wing UAV perform better on these sharp turns because of their flight dynamics are more flexible, fixed-wing UAV can achieve higher flight speed performing the back and forth movements quicker. The point here is that, for large areas and wide turns, FWUAV usually are more suited for the task. For small areas and sharp turns rotary-wing UAV generally perform better [93] [94] [95].

In area coverage with communication, the agents can communicate between themselves and to the base station. The agents are positioned such that the base station, or commonly referred as "depot", and all the agents on the area stay connected during the mission. Each agent senses it own region and sends the data back to the base station and/or to neighbor agents. This method allows surveillance to be persistent since each agent just surveil one specific region. Drawback is that these strategies requires a large number of UAV when compared to other methods [88].

3.2.1 Non Communicative

On non-communicative area coverage problems, comprehensive recent surveys of methods can be found in [96]. On large majority of them, the region of interest is decomposed into cells that must be visited and covered. One mark in this topic is the work of [97], where a method of exact cellular decomposition dedicated to coverage task is made. This method divides the space into convex regions that must be covered by sweeping movements. Also, the cells are modeled in an adjacency graph in which edges represents the existence of a common boundary between the two cells. The area planning in then made by solving first the graph sequence and then sending to the UAV the cells sequence.

In [98] the author adapted the method from [97] to work on smooth and polygonal areas. In this work, instead of searching for edges and vertices like in [97], the algorithms looks for connectivity changes. Further on, [99] used the previous referred method and improved it by describing each cell as narrow or vast, being the behavior of the algorithms slightly different from cell to cell. In these works the only optimization is done on the path sequence. However like referred on last section 3.2, by choosing the correct line sweeping direction we can drastically reduce the energy used on the path. The right angle to perform the line sweeping movement on any polygon area was developed by [17] and [92] goes even further proving that, for UAV, a path with less turns is more efficient in terms of route length, duration and energy. In [100] the target area was partitioned into convex polygons and each one of these polygons was assigned to an UAV. In these cells the path was optimized by using the work of [17] to follow the right angle of line sweeping direction.

Additionally, [101] presented an alternative approach by presenting an area coverage strategy that combines the ideas of cell decomposition of [97] with the optimal line sweeping of [17] and [92]. Thus, the author uses a graph representation where the cells are graph edges, and which the solution consist in the so well known *Traveler Sales Person* (TSP) problem. Where the objective is to minimize the distance that the traveler needs to do visiting every edge at least one time. Off course this work was a perfect synergy with [100] once the use of multi-drones reduces the time of the mission by reducing the workload for each agent and by the capability of drones to compensate the work of others in case of malfunctions or downtimes. Following this vision, a optimized non-communicative area coverage can be considered as a vehicle routing problem, like shown in [102], [103] and [104].

As last, the work on [105] and [106] goes more deeper on multi-UAV path planning using TSP problem resolution philosophy alongside with work [92] and using sensors faced downwards taking into account different resolutions an also temporal coverage of points of interest, modeled as graphs. On [80] the author consider for this kind of coverage the existence of different ground stations which mean different launch points of UAV, where he tries to minimize the longest tour performed by every UAV, resulting in the minimization of total mission time.

UAV technologies, obtain a series of overlapping or non-overlapping aerial images from the ground where these images are usually post-processed for the extraction of desired information. In this context, efficient UAV path planning algorithms are of great importance, since the operation time, costs and the quality of the information extracted from the images are directly related to the quality of such planning.

3.2.2 Communicative

Communicative area coverage, have the same basic principles as the non-communicative area coverage fundamentals. Most of the works where area partitioning, reduced path distance, until a certain waypoint, cells, area attribution for each agent occurs; are from extreme importance and utility on communicative area coverage strategies since the main task is the same as the non-communicative strategy, to surveil a region of interest [89] [88].

However, communications plays a much more important part in the overall operation of an UAV. Communications gives more robust area coverage algorithms. It allows a base station to know the location and state of each agent, being able to obtain images from the observed area in real-time using wireless communication system. This is useful, especially if the data is critical and time-sensitive.

In this type of area coverage the use of MANET offer a way to maintain connectivity between the agents by using radio frequency technologies. This is very useful once it doesn't require any pre-installed infrastructure to operate [88] [107]. With this, UAV nodes can be dynamically located into arbitrary and temporary network topologies. These networks provide the most mobile and least purpose-specific type of wireless and ad hoc networks [108]. They are meant to provide immediate connectivity However, as it uses wireless communication as a medium, characteristics such as high bit error rate, path loss, weak signal strength, delay, interference and penetration loss problems should be considered when using this area coverage strategy [88].

In the common communicative area coverage, the purpose is to place a certain number of nodes such that a region of interest is completely covered and all the nodes and base station can communicate. The nodes placement can be classified into static positioning and dynamic positioning schemes. In static positioning, the placement of the nodes is determined by the base station before a task begins. These UAV nodes will go to the predefined locations when the task begins and will stay there. In dynamic positioning, the placement of each UAV is determined during the task. This is done by algorithms based on current positioning and their neighbors positioning, communication range, sensing range and area of interest. This last strategy has the advantage of agents reposition themselves in case of one or more missing nodes in the region, being fully autonomous.

For this area coverage to be possible, theres a minimum number of agents and for this, works like [109] where the region of interest is subdivided using identical clustering cells with no overlaps between the cells are needed to understand our agent requirements. These clustering shapes are commonly the UAV footprints and can be a circle, square or hexagon. The circle shape is the most natural cluster because coincides with the shape of radio transmission range. Ideal shape for a clustering is the one that tessellating cells are adjacent to each other without any hole or overlapping areas. The shapes with these properties are triangles, squares and hexagons. However sensors, like cameras, cannot have these footprints, resulting in overlapping images or holes in the coverage. In terms of number of agents, for this coverage, it depends mainly on the sensing region of each agent and the area and shape of the region to be covered. As seen in [88] and [17] equal areas may require different number of agents due to region shape.

The works in [110], [94] and [111] propose dynamic communicative area coverage algorithms based on peer-to-peer algorithms just like the equilibrium of molecules denominated by Distributed Self-Spreading Algorithm (DSSA) [110]; based on Voronoi decomposition, denominated as VD-Based deployment Algorithms (VDDA), being the large majority of the researches developed on this field based on this algorithm [94]. There are also other algorithms based on centroid schemes [111].

As summary, area coverage over a planar region has been studied extensively when the sensing patterns of the agents are circular and big advances were made in works [89], [112] and [113]. Despite other sensing patterns like mentioned before.

[114], [115], [116] and [117] works use a based Voronoi or similar partitioning of the region of interest and use a distributed optimization for communicative area coverage. On [118] and [119] authors go further by adding model predictive control, or in [120] by applying game theory to the method. There is also significant work about other sensing patterns like in [109], [121], [122] and [123]. And even other jobs that explore area coverage algorithms avoiding Voronoi partitioning like the ones in [19] [124].

Both convex and non-convex domains for region of interest shape, have been also studied on literature [125] and [126]. Further on, more researches focused on Multi Aerial Vehicle (MAA) need to be highlighted in the context of this work. Like the researches of [127], [128], [129] and [130]. Mostly based on Voronoi tessellations, some developments were achieved with MAA area coverage algorithms when adding cameras to the surveillance technology, just like applied in [131]. Drawback on all of these researches is that, there was no limit to the allowable height that can be reached by an agent, and in the cases where overlap exists, is considered an advantage for covering all the area instead of getting a high quality image of the region of interest.

3.3 Related Works

3.3.1 A decentralized strategy for multi-robot sampling/patrolling: theory and experiments

This work presents a decentralized coordinated strategy for a multi-robot surveillance mission. The surveillance mission in this work is interpreted as a sampling problem (a problem where we want agents to explore or go into certain region and take some data from that specific region). On this work, several realistic constraints and the time/spatial variance of the information were heavily taken into account. Furthermore, the work made use of Voronoi tessellations for decomposing the area to be explored and Gaussian processes to define which regions inside the region of interest were more relevant. Adding to each coordinate on the region an associated importance weight. In the end, each robot, based on local information, defines its next point to visit accordingly to the objectives imposed. Also on this work several numerical simulations and experiments involving a team of robots were mane in order to confirm the proposed method [63].

3.3.2 Area Coverage Algorithms for Multi-agent Surveillance Tasks

In the thesis [88], algorithms for area coverage in non-communicative and communicative surveillance tasks using a team of UAV was applied. Theoretical scenarios were simulated, and in those scenarios, different methods were studied in the context of communicative or non-communicative algorithms. In the non-communicative method, the UAV agents sweep back and forth over a region of interest to cover the area. The polygon which defines the area of interest is decomposed and many different ways are presented. The algorithms for this scenario had one constraint which was that the base station could be only one and the UAV need to start in the base station.

In communicative coverage, methods and algorithms for static node positioning and dynamic node positioning were presented. The algorithms for dynamic node positioning were applied to assist a situation where the nodes are at the predetermined location from static positioning, but there is one or more missing nodes in the region to be covered because of malfunction or just battery dead situations. The results and analysis were then made testing which algorithms boosts the coverage performance [88].

3.3.3 Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time

In the work in [18], a solution for non-communicative area coverage situation is presented with the optimization of time and guarantee of using the minimum number of agents in the shortest time possible. In this work, for solve the problem of area coverage with minimum time coverage of ground areas, a team of drones equipped with sensors and cameras facing down was used, although it has the flexibility to use another kind of sensors. The solution proposed on this work was divided into distinct parts. The first one was to model the task as a graph whose vertices were geographic coordinates determined in such a way that a single UAV would cover the area in minimum time. The second was solving a mixed integer linear programming problem, formulated accordingly to the graph variables defined in the first part, to route the team of UAV over the area.

The main contribution of this work and methodology when compared to primitive researches and solutions for this problem is that this work solves some practical problems encountered only on real tests like the setup time, this is, the time that the operator takes to prepare and calibrate the drone to fly. To demonstrate the contributions made by this methodology simulation and tests are shown to prove that this area coverage algorithms for long areas or small number of agents works and its optimized [18].

3.3.4 Collaborative visual area coverage

In [16] the problem of visual area coverage using a MANET is presented. Each UAV is assumed to be equipped with sensors facing downwards with a conical field of view which covers all points withing a circle on the ground. The diameter of that circle is proportional to the altitude of the UAV, this is, the efficient area covered by each UAV is proportional to its altitude. In this work, a quality function is defined and determines that the coverage quality decreases with altitude. Then, a distributed control law that maximizes a joint coverage-quality criterion by adjusting UAV spatial coordinates was developed. As last in this work, simulations were made to prove the statements and methods mentioned before. This work is a perfect example of a communicative area coverage method [16].

3.4 Conclusion

In this chapter 3, the literature review was made by defining each problem and situation where our thesis supports. The literature review made aims to refer the works developed in the area, forming this way a state of the art of the problem we want to solve. For that, we started by defining what surveillance was and how we could do it with the use of UAV. Then, we shown what area coverage problems are and which solutions are proposed, for specific methods or, just for general area partitioning. Then we explain the work made on last researches about non-communicative area coverage [18], trying to implement a sort of chronological evolution. The same was made for the communicative area coverage [16] strategies. For last, four related works were presented (essentially [18] and [16]). In those works the authors developed methods in which this thesis supports and gets the information necessary to be done. As title of example, in Figs 3.1 and 3.2 examples of non-communicative method and communicative method in a multi-drone concept is showed.

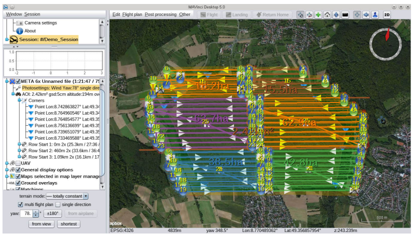


Figure 3.1: Example of Non-Communicative Area Coverage (Mission planner GUI).

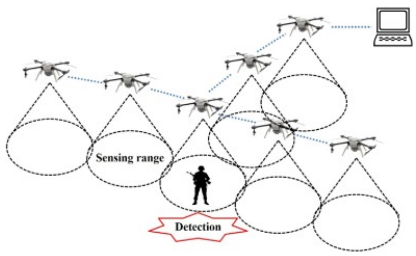


Figure 3.2: Example of communicative Area coverage (MANET).

Chapter 4

Problem Formulation

In the context of surveillance, drones are the disruptive technology that will change in a very near future the way we control our crops, monitor fields or just secure an area. For that to be possible, we still have some challenges to solve in order to release the full potential out of the use of UAV.

One of those challenges is a way to deal with the UAV short flight times, especially for the smaller ones once they are powered by a Li-po battery. The short flight time we have on average drones, approximately 30 min, is a big constraint on area coverage methods and must be taken into account. One solution for this problem is for the drone to detect his battery levels and return to base station autonomously to recharge.

The second challenge is the number of available UAVs. Most of the times due to the lack of budget, or simply the capacity to control a large number of UAVs at the same time, makes it hard to develop a method to surveil the area of interest persistently, accurately and with good quality.

Other challenges we have to face when we want to implement an area coverage algorithm based on multi-UAV cooperative work are the need for a pre-existent infrastructure for communication or the range limitation since we may not have enough drones, operators or battery to cover completely the area of interest.

Another challenge of implementing a way to accurately cover a region of interest is the coverage quality, this is, the image quality of the area we are covering. Commonly the region image definition of what its beneath decreases with height.

The scope of this work is to present methodologies and technologies of surveillance using multi-UAV systems. In order to fulfill that objective, an area coverage problem is taken into account, and different methodologies are applied to solve, depending on the conditions, as best as possible. In an area coverage problem, we can have many different situations such as the region of interest's shape, the area of the region, workload division, tasks, among others.

For this, in this thesis, we will assume a convex area to be surveilled by a team of drones. The objective is then to answer questions such as what maneuver, which UAV system, what methodology and whats the best solution for the asked questions.

Moreover, for each method, the non-communicative and the communicative, different assumptions are made relying on these methods specification. Thus, on coming sections, we will present the problem and assumptions for each of them.

4.1 Non-Communicative

When the number of agents available is not enough to cover the area persistently, this is, constantly, a non-communicative preplanned coverage, using back and forth movements, has to be done. In other words, offline path planning. In that strategy, it is assumed that a group of N drones have the mission to cover, as quickly as possible, a polygonal convex area represented by P vertices in $\Re^2$. If a non-convex area is to be covered, it is assumed that P is the convex hull of the area of interest.

In these assumptions, we want to make the path quickly as possible because if we do this path systematically we will be able to monitor each area more times per time frame. Also, the UAV can be a FWUAV or a rotary-wing UAV. Although the FWUAV perform better for long distances because they usually achieve higher speeds, they are less flexible in turns like explained in Chapter 3. However as long as the turns are wide and not sharp, FWUAV perfect suit this kind of maneuver [93] [94] [95].

Also, it is considered that the maximum flight time for each drone is finite and well known. Each drone is equipped with an on board camera facing downwards. The mission of this drones on this coverage is to sense the entire region specified by P. The flight altitude is constant through all mission so the observation of the region of interest is uniform and the battery consumption is minimized. We also assume that all drones are homogeneous.

Related to the operation, we assume the necessity of a setup time before each drone set flight. This setup time is referred to the time wasted by the operators to prepare the drone while the mission is executed. This assumption is very useful once it is an issue that occurs in every practical experiments when launching a team of drones to set flight.

Given this, the specific problem we are dealing in this method is divided into two parts. In the first part, we want to discover the number of drones required to perform the coverage in minimum time. This happens because if we just have one operator to launch several drones, their setup times stack and the task may not be optimized. So, we want to discover the number $n \leq N$ that minimizes the objective function, minimizing the time taken to cover the area represented by P.

In the second part, we need to share the coverage jobs among the drones, this is, specify the paths so that the mission is done in minimum time. The fact that we have setup times and the drones are not launched all at the same time unless we have one operator for each drone, makes the work not being divided equally among all drones. The solution for this method problem is presented in Chapter 5 [18].

4.2 Communicative

Given a certain area to cover, and preferentially enough drones to cover all the area, a communicative area coverage method can be applied, relying on drones swarm behavior for a real-time full coverage. If the number is not enough and we still want to implement this method the percentage of covered area will decrease.

In this method, we assume A as a convex polygonal area in $\Re^2$ requiring surveillance. We also assume that the group of drones is already positioned in the area, excluding the job of making them going until there. Inside the area of interest, each drone is characterized by a spatial coordinate defined as $X_i = [x_i y_i z_i]^T, i \in I_n$ where $I_n = \{1, \cdots, n\}$. We also consider the vector $f_i = [x_i y_i]^T, f_i \in A$ to simulate the footprint of the sensor of each drone over the planar region. The altitude is not constant once in this coverage we aim not only to cover the area but also to have coverage quality. This implies, a band of altitudes defined as z_i^{min} and z_i^{max}, being $z_i^{min} < z_i^{max}$ and $z_i \in [z_i^{min}, z_i^{max}], i \in I_n$, where z_i is the altitude of each considered drone. It is assumed that $z_i^{min} > 0, \forall i \in I_n$ in order to avoid crashes on the ground. If the field has obstacles they need to be known in advance so the minimum altitude is higher than the obstacle height. It is also assumed that the maximum altitude is limited by maximum communication range to the station. Note that minimum altitude and maximum altitude are the same for every drone in this coverage method.

On this problem, we also assume simplified dynamic models, being each drone approximated as a point of mass, resulting in the kinodynamic model

$$\dot{f}_i = t_{i,f} \qquad f_i \in A \qquad t_{i,f} \in \Re^2 \tag{4.1}$$

$$\dot{z}_i = t_{i,z} \quad z_i \in [z_i^{min}, z_i^{max}] \quad t_{i,z} \in \Re \tag{4.2}$$

where $[t_{i,f}, t_{i,z}]$ is the 'thrust' control input corresponding for each drone, in this case, modeled as node.

Identical to the offline path planning method in the previous section, the drones are considered homogeneous and with sensors, in this case, cameras, pointing downwards, being the camera sensing pattern considered conic despite can have other patterns (usually cameras have square footprints as it is gonna be seen in some experiments). Thus, the region of A sensed by each drone is a disk (in the case of a conic pattern) defined as

$$C_i^s(X_i, a) = \{f \in A : ||f - f_i|| \leq z_i \cdot tan(a)\} \quad i \in I_n \tag{4.3}$$

where a is half of the angle of the sensing cone. In Eq. 4.3 we can see that the covered sensing region are all the points within the radius of $z_i \cdot \tan(a)$, being the covered sensing region directly proportional to the altitude of the drone and the angle of vision of the camera used. We can also see in Fig. 4.1 that the higher the altitude of a drone the larger is the area from A surveyed by the drone's camera.

However as said before, in this coverage method we also aim for image quality, that way we assume that each drone has a quality coverage function $q(z_i) : [z^{min}, z^{max}] \rightarrow [0, 1]$ which is dependent on the drones altitude constraints. The coverage quality of drone i is assumed to be uniform throughout its sensing region C_i^s. The higher the value of $q(z_i)$, the better coverage quality.

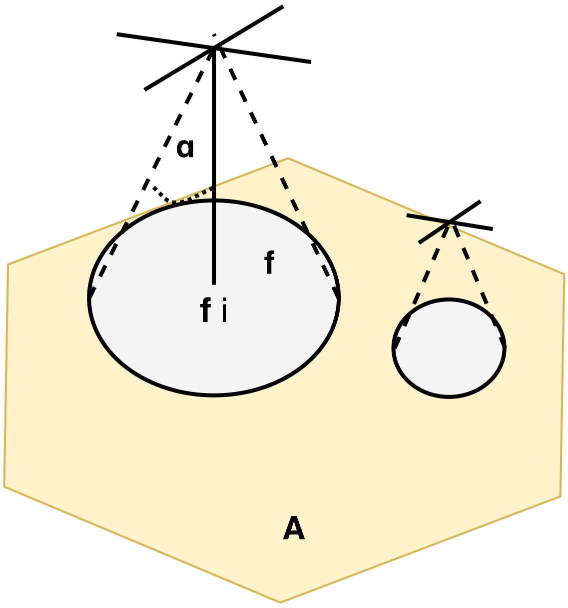


Figure 4.1: Multi-drone area coverage concept [16].

It is assumed as common sense that the higher we are the worst is the image quality we have, since each pixel will represent a larger distance than when at lower altitudes. This coverage quality will be further explained with more detail on Chapter 5.

For last, it is assumed that each point $f \in A$ has an importance weight assigned via a space density function $\phi : A \Rightarrow \Re^{+}$, encapsulating any a priori information regarding the region of interest. Resulting in a final quality objective function designed as O, where

$$O \equiv \int_A max_{i \in I_n} q(z_i)\phi(f)df. \tag{4.4}$$

The solution for this problem is further explained on Chapter 5, at the Methodology.

4.3 Conclusion

In the first problem, where it is assumed that the drones are not enough or the area is to large to be covered in real-time, we want to design a control strategy (under the assumptions mentioned above) such that the route assignment for each drone leads to the minimization of total time taken to cover the area.

In the second problem, where it is assumed that the drones are enough to cover the area, we want to design a distributed system where area covered and quality of that coverage itself are maximized through a joint coverage quality criterion.

Chapter 5

Methodology

On this chapter we will provide our methodology to solve the problems defined in Chapter 4 which are coverage problems using multi-drones with different approach and methods. From the works of [18], [16], [88], [89], and all the other researches highlighted in Literature in Chapter 3, our strategy for solving the problem of covering the area using a group of drones was split in two different methods.

With a non-communicative offline strategy we first decompose the area to see what is the optimized path direction to execute the line sweeping movements, then we transform the solution of the paths creation into a TSP problem, which is a vehicle routing problem, and then we see what is the path for each drone that minimize the total time of the mission, that is, minimize the objective function. As last, we expose how the algorithm implementation works.

With the communicative collaborative visual area coverage strategy we divide the problem into several issues. First we address the coverage quality function and how it works, then we expose how the area of interest is partitioned, in this case with no predetermined cells. Then we explain the distribution of the region of interest for each drone and the cooperative behavior of all agents. For last, we show how we maintain altitude control and stability and how the global algorithm is implemented.

5.1 Non-communicative Strategy

As said before, the strategy adopted to solve problem proposed in Chapter 4 section 4.1 is divided into two parts. In the first part, we decompose the area to be covered as a group of rows similarly like the method proposed in [17]. These rows will be posteriorly be modeled as edges of a graph for solving a mixed integer linear problem, called TSP. From there we obtain the best routes for each drone to follow in order to minimize the time of the operation.

5.1.1 Area Search

We assume that in this type of coverage the drones will fly over the region of interest executing back and forth movements, like a lawnmower, in rows perpendicular to a given sweep direction as shown in Fig. 5.1. While following the rows, the drone is leveled, this is, the camera is pointing down and the image is good, however at the end of the row, when making the curve to return to the area, the camera is not pointing down. Furthermore, as explained by [17], the amount of turns is directly related to the time of coverage of the region. Therefore as suggested by the same author, the first step of the area decomposition method is to find the optimal direction of coverage, which is perpendicular to the smallest height of the polygonal area. In this direction, the area can be covered with a smaller number of rows, thus with a minimum amount of turns. This behavior can also be observed in Fig. 5.1. As proven in [92] and on Chapter 3, a path with less curves is a more efficient path in terms of route length, duration and energy. Note that this rule only apply when the wind is negligible.

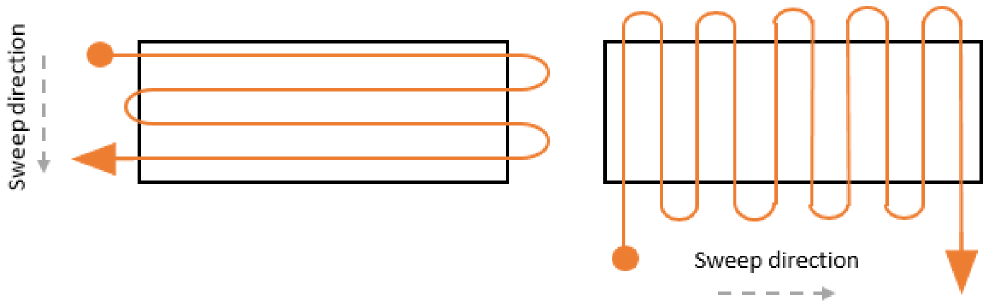


Figure 5.1: Coverage strategy applied on this method of area coverage. In this case, a rectangular area is covered by the use of a back and forth movement along lines perpendicular to the sweep direction. Notice that the number of turns made by the drone in while covering the area is directly related to the sweep direction. Right movement (left figure) reduces drastically the time once the turns are where the energy consumption and time taken is higher [17].

In order to find the optimal direction of coverage for any given polygon, as [17] suggested, we need to make a search on the polygon (area to surveil) and rotate him over a surface in order to check when its height is smaller. In Fig. 5.2 we can see better that how rotating a certain polygon over a surface affects the diameter function. In this case, we want the diameter function to be as low as possible, which means that we want the height of the polygon to be reduced to its minimum [132].

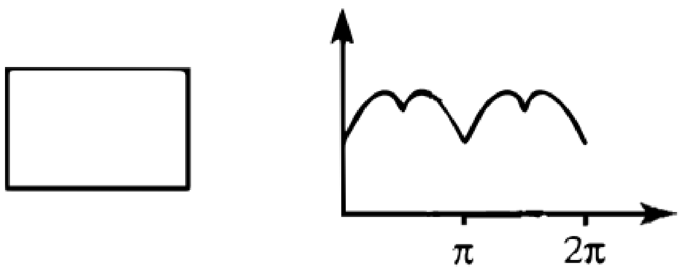


Figure 5.2: Example of diameter function in a rectangle polygon from [17].

Note that to create this diameter function we need to focus in one fixed point that could be for example one vertice, then the size of the edge that connect that point to the rotation surface will represent the height of the diameter function. As shown in Fig. 5.3, the polygon is rotated over a surface, and its height is measured. The best orientation is the one that yields the smallest height, this is, the minimum of the diameter function. This minimum height is represented by h_{min}.

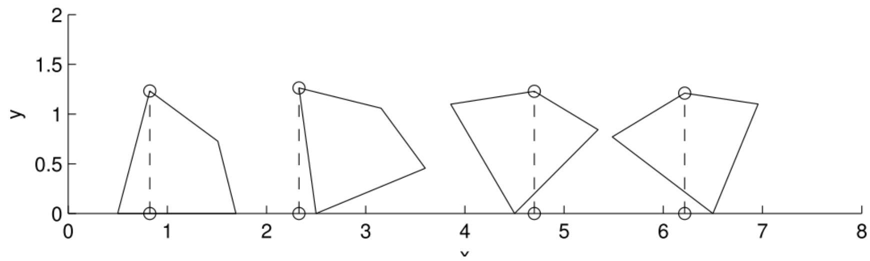


Figure 5.3: Search for the optimal direction of coverage. For that, we need to find the smallest height for the polygon, which means the minimum diameter accordingly to [17] from [18].

Once the optimal sweep direction is found, this means, after the optimal angle of rotation is found is possible to distribute the rows over the area to cover. The rows will have an inclination perpendicular to the optimal sweep direction angle like previously mentioned in this section.

The distance between each row has to be enough to cover all the area without any holes or gaps in the coverage. As we defined on Section 4.1, in our approach the distance between two rows is chosen as a function of the footprint of the on-board camera on the ground. Assuming that the drone camera is leveled with the ground, we can obtain the distance between the rows based on the width of the image sensor, represented per l, the focal distance of the camera's lens, represented per f, both in millimeters which is the most used unit for camera's characteristics. More, the distance between the camera and the ground, represented per H (flying height), in meters, it is possible to compute the width of the coverage area covered by the drones sensor, which is the camera's footprint on the region of interest, represented per L, in meters as well as [18]:

$$L = H\frac{l}{f} \tag{5.1}$$

To understand, how we get Eq. 5.1 we have Fig. 5.4, where the variables referred before are represented and the interaction between them as well.

Thus, the number of rows needed to cover the area is equal to Eq. 5.2 and the distance, in meters, between two rows is on Eq. 5.3. Where h_{min} stands for the polygon minimum height (lowest diameter function value) and, where s represents the percentage of image overlapping between the rows which means that $s \in [0,1]$ where 0 is the limit where image overlapping is 0% and where 1 represent 100% of overlapping images, which is in practical useless in our approach. In our case, minimum overlap reduces the number of rows, decreasing coverage time, covering redundancy and being more efficient in terms of energy consumption.

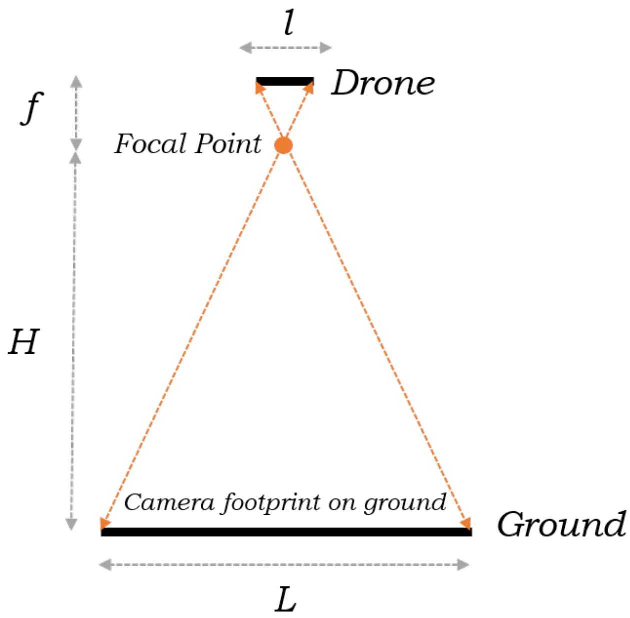


Figure 5.4: Relation between the drone camera and the area covered by that drone when leveled with the ground from [18].

$$R_l = [\frac{h_{min}}{L(1-s)}] \tag{5.2}$$

$$d_l = \frac{h_{min}}{R_l} \tag{5.3}$$

Knowing in advance drone camera specification, desired lane overlap, flight altitude and area of interest to cover we can determine optimal sweep direction and the number of rows. But we still have to plot those rows with coordinates in order to sent flight instructions for each drone. For that, we assume that the polygon that represents the area to be covered is rotated in a way that the optimal direction of coverage is parallel to the x axis of the global reference frame, being the coverage rows defined by two planar points (x,y) with identical y coordinates given by:

$$y_i = i \times d_l - \frac{d_l}{2} \quad i = 1,...,R_l \tag{5.4}$$

and x coordinates defined by the points where the horizontal straight line with coordinate y_i intercepts the borders of the area to be covered. Once the points are computed, they are rotated back to the original orientation. This way we can simplify the application of coordinates to the rows, so drones can receive the coordinates to follow. If desired this coordinates can use an own metric system or GPS if thats the case.

5.1.2 Formulation of TSP problem

Once obtained the coordinates of the rows, we transform the back and forth movement problem into TSP problem in order to see whats the best path to follow by each drone. For that, the extremity points of the coverage rows, along with the coordinates of the base station (place where the operator will launch the drones and also defined in advance), which can also be called depot, are modeled to be a set of nodes V of a graph $G = (V, E)$ where V are the rows coordinates and base coordinates and E are the edges that represent the path between those points. Each node of the graph G is then numbered in order such the number 1 is the base node and the others are numbered in row order as shown in Fig. 5.5. At the end, each coverage row is associated with subsequent even and odd nodes. The edges E in graph G represent all the possible routes between the rows and base, this is all the lines connecting the nodes, creating this way the TSP problem. Each edge is a path, a set of edges is a route. In Fig. 5.5 we can see an example of the full graph representing the coverage problem [18].

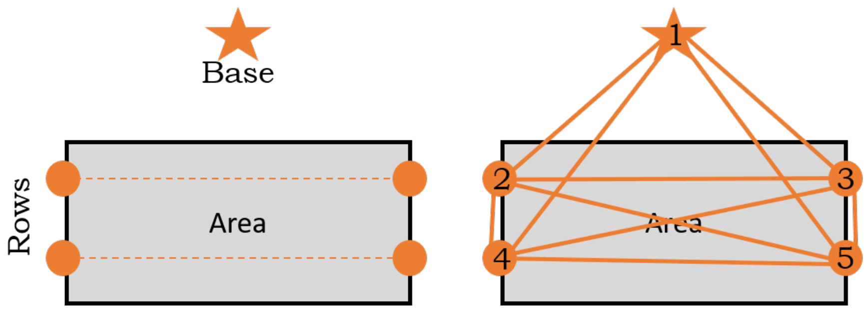


Figure 5.5: On the left, theres the polygon that represents the area to be covered, the start represent the base or depot, and the dots in the area represent the rows that need to be followed in order to patrol all the area. On the right, we have a transformation of the coverage problem on the left side to a graph of a TSP problem where the nodes represent the base and the interception of the rows with the area and the edges represent all the path between nodes. From [18].

Now we just have to write this problem in such way we can solve him. The mathematic representation of this graph G is an $i \times j$ cost matrix C whose elements, $C_{i,j}$, are given by the Euclidean distance between the spatial coordinates of nodes i and j, being this way modeled for any coordinate system, for example, GPS coordinates. Its important to highlight that C is time invariant, symmetric and its elements satisfy the triangular inequality [18].

5.1.3 Path Attribution Strategy

Once the graph associated with the region to be covered is created, the problem coverage can be treated as a TSP problem. In this class of problems, a set of customers must be visited by a set of "sellers", in this case, a set of points to cover need to be visited by drones. For that, the graph nodes will be the points to be visited and the edges the weight associated paths for the drones to reach those points. Without loosing the focus of the real coverage problem we need to define the TSP problem objective function and constraints such that the solution of the TSP problem is actually the solution for this area coverage strategy [18]. The desired solution is a set of routes for drones to follow in order to minimize the total time of the mission. Allowing this way more surveillance per time frame. On a classic TSP problem which is a NP-hard mixed integer linear problem, before we compute to obtain the optimal solution we need firstly to define, the variables, secondly the objective function and lastly the problem constraints itself.

The variables and constants, that constitute this mathematical formulation of the coverage problem are [18] [133]:

- Variable $C_{i,j}$ is the cost matrix of the path between the nodes i and j. In this case, the cost is measured as the distance between node i and j.
- Variable $X^k_{i,j}$ is the binary variable with $X^k_{i,j} \in \{0,1\}$ that indicates whether or not the drone k is going to fly between the nodes i and j.
- Constant $S^k_{i,j}$ is the flight speed of the drone k between the nodes i and j.
- Constant $t_s \in \Re$ is the the setup time. This is, the time that an operator takes to prepare the drone to set flight.
- Constant L_k is the battery duration of drone k given in advance.
- Constant $n \in \aleph$ is the number of drones used in the mission.
- Constant $N \in \aleph$ is the number of drones available for the mission.
- Constant $O \in \aleph$ is the number of drone operators.
- Constant $V \in \aleph$ is the number of nodes (or vertices) of the graph.
- As last, constant $d_k \in \aleph$ represent the extra time necessary to launch the drone k.

Based on the variables and constant defined in the last itemization, the time spent by each drone k to fly its route is mathematically represented by:

$$T_k = \sum_{i=1}^{V} \sum_{j=1}^{V} \frac{C_{ij}}{S^k_{ij}} X^k_{ij} + d_k \tag{5.5}$$

As, in our coverage problem, we want to minimize the total time of the mission, we need to minimize the time of the longest route among the routes of all drones to accomplish the objective. This way, our problem transforms into a min-max problem in which we want to minimize the maximum of T_k. The only way to transform our min-max problem into a linear problem, is by introducing an extra variable λ, which represents the longest drone route. The basic optimization problem, which is the Objective function is then written as [18]:

$$min(\lambda) \tag{5.6}$$

subject to

$$\sum_{i=1}^{V}\sum_{j=1}^{V}\frac{C_{ij}}{S_{ij}^{k}}X_{ij}^{k}+d_k \leq \lambda \quad k=1,...,N \tag{5.7}$$

where,

$$t_s[\frac{k}{O}]\sum_{j=1}^{V}X_{1j}^{k}=d_k \quad k=1,...,N. \tag{5.8}$$

As shown previously, constraint in Eq. 5.7 accounts for the individual cost of drone k. This corresponds to T_k. By defining this constraint along with the objective function in Eq. 5.6 using variable λ, we are acting as a linear version of the min-max problem, where the maximum cost among the ones of all drones must be minimum. By doing this, we are, in practice, minimizing the time taken to cover the complete area of interest [18].

In Eq. 5.7 and 5.8 the term d_k accounts for the setup time which is an extra cost (time) that corresponds to the time spent by an operator O to prepare and launch the drone. This is an important feature once its one of the main contribution of this algorithm proposed by [18] because, if we only have one operator for all the team of drones, each drone will have the extra time d_k which will be cumulative. For example, if we have two drones but only one operator, while the first drone is being prepared to be launched the second drone is doing nothing, waiting to be launched. Note that Eq. 5.8 ,cover all possibilities and that d_k is null if the drone is not used, once X_{1j}^{k} is zero, meaning that drone k did not leave the base station at all (represented as the node 1 like referred previously). Further examples and exploration of the constraint imposed by the Eq. 5.8 can be seen in literature [18]. Notice that this constraint is also why all the M drones may not be necessary in order to obtain the best solution because they have the extra time of being set up. In the case, we have as many operators as drones, or simply the technology allows drones to set flight autonomously once they receive instructions, then they wont have t_s or it will be negligible and the function will directly vary to the number of drones.

To complete the optimization problem and to successfully solve the offline non-communicative coverage problem presented in Section 4.1 through a TSP problem we need to incorporate other constraint into the basic problem. The constraints are defined as:

$$\sum_{i=1}^{V}\sum_{j=1}^{V}\frac{C_{ij}}{S_{ij}^{k}}X_{ij}^{k} \leq L_k \quad k=1,...,N \tag{5.9}$$

$$\sum_{k=1}^{N}\sum_{i=1}^{V}X_{ij}^{k}=1 \quad j=2,3,4,...,V \tag{5.10}$$

$$\sum_{i=1}^{V} X_{ip}^{k} - \sum_{j=1}^{V} X_{pj}^{k} = 0 \quad p = 1,2,3,...,V \quad k = 1,2,3,...,N \tag{5.11}$$

$$u_i - u_j + V \sum_{k=1}^{N} X_{ij}^{k} \leq V - 1 \quad i,j = 2,3,4,...,V \quad u_{i,j} \in Z \tag{5.12}$$

$$\sum_{k=1}^{N} X_{i,i+1}^{k} + \sum_{k=1}^{N} X_{i+1,i}^{k} = 1 \quad i = 2,4,6,...,V \tag{5.13}$$

$$\sum_{k=1}^{N} X_{i,i+1}^{k} = \sum_{k=1}^{N} \sum_{j=\{1,3,..\}\setminus\{i+1\}}^{V} X_{i+1,j}^{k} \quad i = 2,4,6,...,V \tag{5.14}$$

$$\sum_{k=1}^{N} X_{i,i-1}^{k} = \sum_{k=1}^{N} \sum_{j=\{1\}\cup\{2,4,...\}\setminus\{i-1\}}^{V} X_{i-1,j}^{k} \quad i = 3,5,7,...,V \tag{5.15}$$

$$\sum_{k=1}^{N} \sum_{j=1}^{V} X_{1j}^{k} = n \tag{5.16}$$

$$n \leq N \tag{5.17}$$

Eq. 5.9 limits the maximum time of flight of drone k by its battery duration L_k. We neglect in this method that the drone's battery decreases while is being set to fly and that the the charge decreasing does not happen. This way, the time of flight for each drone is simply $T_k - d_l$, accordingly to Eq. 5.7. It is important to alert that the constraint devised in Eq. 5.9 can make the problem infeasible once the battery might not be enough if the area is to big for the drone speed and time flight time to short. Being in this case, the only solution available to increase the number of agents. However, there are situations where the methodology has to be changed from the root in order to have a solution. One less practical solution is to simply acquire better drones.

Constrain in Eq. 5.10 enforces that each node, in exception of the base (node 1), is visited by only one drone, avoiding redundancy. On the other hand, constraint in Eq. 5.11 guarantee, that the drone that arrives at a given node is the same one that leaves the node. Otherwise, the coverage would simply not work.

To enforce that each drone path starts and ends at the depot (node 1) and to eliminate internal cycles, a standard sub-tour elimination constraint based in [134] is represented in Eq. 5.12.

The constraint in Eq. 5.13 was designed to make sure that the TSP problem will make the drone cover the area modeled as the graph G. On this equation, each drone, having visited one of the nodes of a coverage row, must visit the other node of that row. This is possible to be designed and expressed like that due to the way we numbered the nodes. We can clearly understand it if we give a look at the graph example present in Fig. 5.5. The numeration scheme of nodes applied in Eq. 5.13 forces the drone that visit an even node also visits the next odd node. Furthermore, a drone that visits an odd node must visit the previous even node. Thus, this is essential to make the TSP solution an actual area coverage solution.

In order to avoid drones crossing areas following edges that are not parallel to the coverage rows, two constraints are added to the optimization problem. These constraints are present in Eq. 5.14 and Eq. 5.15. On these equations, the drone avoids executing sharp turns and also photos from being taken in different directions.

In Eq. 5.16 and 5.17 in opposition to [134] guarantee that the number of drones is the optimal, this is, the number of drones is chosen accordingly to the minimization of λ [18].

It is expected that computing the objective function 5.6 and constrains in equations 5.7-5.17 would give us the solution where the minimum number of drones would follow the shortest possible paths, covering in minimum time the area of interest. However, although the mission time will indeed be minimum, the objective function does not explicitly take into account the number of drones, which may cause the computed solution to consider a number of drones that its not optimal. Moreover, our problem minimizes the longest path in terms of mission time, once the best path is found, there is no way to be sure that the paths for the other drones are optimized. To make sure that the optimal solution is absolute and not the local optimal solution one strategy is devised [18]. The strategy to generate an optimal global solution for all drones paths and number of drones used, independently of parameter choices, is using an iterative solution that consists in solving the problem more than once. On the first iteration, the original graph is used and the problem is solved as previously. Then, on the second iteration, we remove the longest path which is already optimized and optimize the second biggest path and do it recursively until all the nodes are removed (except the base). Obtaining this way the best solution possible for the area coverage throughout this method. It is also guaranteed that with this solution, our primary objective is optimal which is to minimize the cost of the route of the highest cost.

In chapter 6 simulations will be made in order to support this method effectiveness, mainly developed by [18] and other related literature.

5.1.4 Algorithm

In terms of algorithm, this method requires firstly the user to input several data as the drone flight time, flight speed, flight altitude, number of operators, number of drones and their setup time as well. The user also needs to add the camera specifications, the lens specifications (to define the FOV) and image overlap (which implied lanes width). Then, the user sets the coordinates that define the area to cover through the vertices coordinates of the area (polygon). After that, the computer calculates lanes width and the optimal sweep direction. Once these calculations are done, rows are defined. Once the rows are defined, the nodes that are the points from the rows that intercept the border of the region to be surveilled are modeled as nodes of a graph alongside with the depot point. Transforming itself as a graph. Then the algorithm executes all the possible solutions in order to find the optimal solution. Once it finds the optimal solution, it removes the nodes bellowing to that the optimal drone path and executes again, iteratively until there are no nodes on the graph left in exception of the base station. In that point, the algorithm stops and the best solution is plotted showing the optimal path for each drone.

Summary of this algorithm is briefly explained on algorithm overview 1, where the variables notation is the one exposed previously in this section.

Algorithm 1: Non-communicative area coverage method algorithm

Output: Route coordinates for each drone
Result: Global optimal solution of the problem
Data: Drones specifications, number and operators number
Input: Vertices coordinates of the area to be surveilled
begin
    $Operators \longleftarrow O$; $AgentNumber \longleftarrow N$; $SetupTime \longleftarrow t_s$
    $FlightSpeed \longleftarrow S^k$; $FlightAltitude \longleftarrow H$; $Camera \longleftarrow l, f$; $Overlap \longleftarrow s$
    Initialization
    $Base \longleftarrow base(x,y)$
    $Area \longleftarrow PolygonVertices(x,y)$
    for *Area* **do**
        Plot the Area on the map
        Find Optimal sweep direction
        Rotate polygon
        Determine:
        $Numberofrows \longleftarrow R_l$
        $RowsDistance \longleftarrow d_l$
        Obtain rows coordinates from intersection with area border
    end
    Model nodes and base station as graph $G = (V,E)$
    while *nodes > 1* **do**
        Search for the minimization of the max of 5.6
        Execute constraints: from 5.7 to 5.17
        if *optimal Found* **then**
            Remove nodes bellowing to the minimization of the longest path time
        else
            continue
        end
    end
    Plot the optimized paths for each drone to cover the area
end

The algorithm was implemented in MATLAB, using YALMIP toolbox and GUROBI optimizer for faster and accurate optimization.

5.2 Communicative Strategy

Communicative Strategy for area coverage problem method applied on this thesis is mainly the procedure followed by [16] and [19]. In this coverage method, we want to a given region, satisfy two different conditions, one is the area covered and the other one is the quality of that area covered. Drawback of this method is that due to the fact we want to receive in real-time a good image quality makes the area not being covered in 100% (in cases where there are enough drones for all the area) because theres the image quality factor, which is a function dependent from the height of each one of the drones like shown in Eq. 4.3.

In this second method we will explain how the coverage quality function is implemented, then we explain how the space is covered and sensed by each agent. Posteriorly, we will explain how they interact with each other to manage the space covered and how they collaborate for the benefit of the overall performance.

Also, theres the possibility to adapt the presented method with the work of [19], in order to see if we consider also in the problem, the situation where the sensed region by the camera is not equal in all directions and, that way, controlling the direction of each drone may also increase the system performance. For last, we explain the method to maintain the altitude stability off all the drones. And then, we expose the summarized algorithm for this method.

5.2.1 Coverage quality function

Bearing in mind the problem and assumption made in section 4.2, and the equations expressed in it, like the quality function, an uniform quality throughout the sensed region C_i^s can be used to model the sensors, in this case, cameras facing downwards like in [18] [88] [133], that provide uniform image quality. Accordingly to method of applied in [16] the uniform coverage quality function $q(z_i)$ was chosen to be:

$$q(z_i) \quad = \quad \begin{cases} \frac{((z_i - z^{min})^2) - (z^{max} - z^{min})^2)^2}{(z^{max} - z^{min})^4} & f \in C_i^s \\ 0, & f \notin C_i^s \end{cases} \tag{5.18}$$

This function is designed in such way that $q(z^{min}) = 1$ and $q(z^{max}) = 0$ which makes sense since when closer as possible to the ground the image is obviously better in terms of definition that when the drone is at big altitudes, where it reaches a point that definition is terrible and the quality function is set to be 0. Furthermore, $\frac{\partial q(z_i)}{\partial z_i}$ exist within C_i^s, which is a property that will be needed when deriving the control law for the coordination method presented in further sections.

From section 4.2 its defined that function $q(z_i)$ as an interval of $z_i \in [z^{min}, z^{max}] \to q(z_i) \in [0, 1]$. This way, the derivative $\frac{\partial q(z_i)}{\partial z_i} : [z^{min}, z^{max}] \to [q_d^{min}, 0]$ is assess as $q_d(z_i) \equiv \frac{\partial q(z_i)}{\partial z_i}$ where:

$$q_d(z_i) \quad = \quad \begin{cases} \frac{4(z_i - z^{min})[(z_i - z^{min}) - (z^{max} - z^{min})^2)^2]}{(z^{max} - z^{min})^4} & f \in C_i^s \\ 0, & f \notin C_i^s \end{cases} \tag{5.19}$$

Also, resulting from Eq. 5.19, $q_d^{min} = q_d(z^{min} + \frac{\sqrt{3}}{3}[z^{max} - z^{min}] = -\frac{8\sqrt{3}}{9(z^{max}-z^{min})}$. The Eq. 5.18 and 5.19 plots are shown in Fig. 5.6. In this Figure, we see how the coverage quality function decreases as higher the drone is, on the left, representing the uniform coverage quality function, and how this function variates in order to z_i on the derivate plot, on the right side. It is possible to clearly understand that for the minimum height the quality is 1 and that for maximum height the quality is 0 [16].

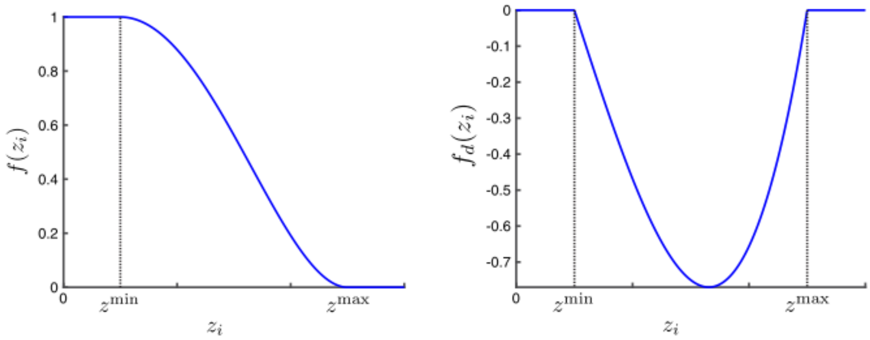


Figure 5.6: Function plot of uniform coverage quality function ($q(z_i)$) [Left] and plot of the derivative of $q(z_i)$ ($q_d(z_i)$) [16].

5.2.2 Space partitioning

The distribution of the area for each drone to surveil in this model is not as the majority of area coverage method with multi agent systems. In this method of [16] the assignment of responsibility to the drones is made in a process close to [19], where only the subset A sensed by the drones is partitioned. In this method, to each drone is assigned one cell from the area sensed by the cameras and its defined as W_i where [19]:

$$W_i \equiv \{f \in A : q(z_i) \geq q(z_j) \quad j \neq i\} \tag{5.20}$$

With the equality expressed in Eq. 5.20 being true only at the boundary ∂W_i, so that the cells W_i comprise a complete tessellation of the sensed region. The only trouble with this method is that with the shape of the area A and the sensor footprint on the planar region we may not achieve 100% of area covered.

Because the coverage quality function is uniform, $\partial W_i \cap \partial W_j$ is either an arc of the ∂C_i if $z_i < z_j$ or of ∂C_j if $z_i > z_j$. If happens that $z_i = z_j, \partial W_i \cap \partial W_j$, is then, chosen arbitrarily as the line segment defined by the two intersection points of ∂C_i and ∂C_j as result, the cells consist of circular arcs and line segments. In literature [19] and [16] examples of the different tessellations are exposed in Fig. 5.7 and 5.8. In these figures, we can see the space partition in many different situation when following the scheme defined in Eq. 5.20.

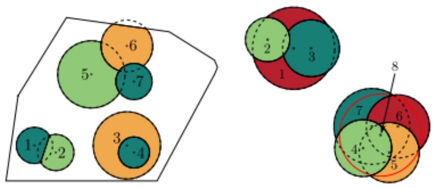


Figure 5.7: Examples of space partitioning examples from [16].

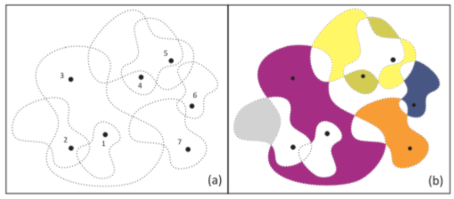


Figure 5.8: Examples of space partitioning examples from [19].

Following this partitioning scheme, the network's coverage performance can be written as:

$$O = \sum_{i \in I_n} \int_{w_i} q(z_i)\phi(f)df. \tag{5.21}$$

For Eq. 5.21 to be true we assumed and defined that neighbors, represented by N_i of node i are:

$$N_i \equiv \{i \neq j : C_i^s \cap C_j^s \neq \emptyset\} \tag{5.22}$$

The neighbors of drone i are those drones that sense at least a part of the region that drone i senses. It is clear that, due to the partitioning scheme used, only the nodes in N_i need to be considered when creating cells W_i. It is also important to highlight that, the partitioning scheme used is not a complete tessellation of the area A but of the subset of A, represented by P, that corresponds to the sensed area by the drones cameras, $P = A \setminus \bigcup_{i \in I_n} W_i$. The cells resulting from the space partitioning are compact. However, they are not always convex [16] [17].

5.2.3 Spatial Coordination

The coordination between the drones is what makes this coverage method so interesting. Based on the different equations aforementioned it is possible to create a spatially distributed coordination algorithm that allows the communicative area coverage to happen. In this algorithm, we search for the optimization of two aspects, the area covered, which is the area sensed, and, the quality of that sensed area. Thus, based on drones kynodynamic model defined in Eq. 4.1 and Eq. 4.2 and their sensing performance represented in the Eq. 4.3 and in the coverage criterion presented in Eq. 5.21 a gradient based control law is defined. This control law utilizes the partitioning scheme on Eq. 5.20 and result in a monotonous increase of the covered area. Leading for the improvement of both, image quality, and area covered.

5.2.3.1 Control laws

In order to reach the right control laws, we initially evaluate the time derivative of the optimization criterion O [16] [19]

$$\frac{dO}{dt} = \sum_{i \in I_n} [\frac{\partial O}{\partial f_i} \dot{f_i} + \frac{\partial O}{\partial z_i} \dot{z_i}] = [\frac{\partial O}{\partial f_i} u_{i,f} + \frac{\partial O}{\partial z_i} u_{i,z}] \tag{5.23}$$

The we make use of the gradient based control law in the form of

$$u_{i,f} = \alpha_{i,f} \frac{\partial O}{\partial f_i} \tag{5.24}$$

$$u_{i,z} = \alpha_{i,z} \frac{\partial O}{\partial z_i} \tag{5.25}$$

Resulting in a monotonous increase of O where $\alpha_{i,f}$ and $\alpha_{i,f}$ are positive constants that represent the control law gain.

Continuing to follow the method of [16] and [19], by using Leibniz integral rule we get

$$\begin{aligned} \frac{\partial O}{\partial f_i} = \int_{\partial W_i} v_i^i n_i q(z_i) \phi(f) df + \int_{W_i} \frac{\partial q(z_i)}{\partial f_i} \phi(f) df + \\ \sum_{j \neq i} [\int_{\partial W_j} v_j^i n_j q(z_j) \phi(f) df + \int_{W_j} \frac{\partial q(z_j)}{\partial f_i} \phi(f) df] \end{aligned} \tag{5.26}$$

where, v_j^i stands for the Jacobian matrix with respect to f_i of points $f \in \partial W_j$, such that

$$v_j^i(f) \equiv \frac{\partial f}{\partial f_i}, \quad f \in \partial W_j, \quad i, j \in I_n. \tag{5.27}$$

Knowing that $\frac{\partial q(z_i)}{\partial f_i} = \frac{\partial q(z_j)}{\partial f_j} = 0$ we then, obtain

$$\frac{\partial O}{\partial f_i} = \int_{\partial W_i} v_i^i n_i q(z_i) \phi(f) df + \sum_{j \neq i} \int_{\partial W_j} v_j^i n_j q(z_j) \phi(f) df. \tag{5.28}$$

Whose two terms address how the movement of a drone i affects the boundary of its cell and the boundaries of the cells of other drones. It is clear that only the cells W_j which have a common boundary with W_i will be affected.

As shown in the Fig. 5.9 the boundary of ∂W_i is decomposed into a set of boundaries representing each one the intersection with the other boundaries. Resulting that, the boundary of ∂W_i is the reunion of the intersection of the boundary ∂W_i with all the others.

Since we assume that the region of interest is static, and, only the common boundary $\partial W_j \cap \partial W_i$ of drone i with any other drone j is affected by the movement of node i then, $\frac{\partial O}{\partial f_i}$ is simplified. Also, accordingly to [19] and [16], the boundary $\partial W_j \cap \partial W_i$ is common among nodes i and j, it

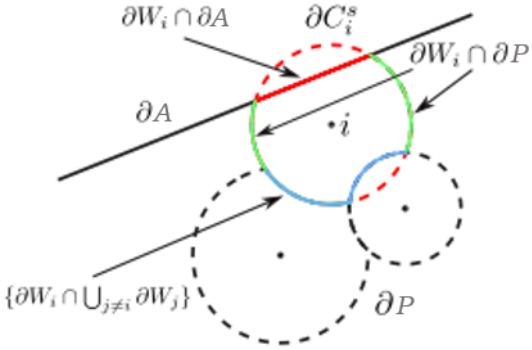


Figure 5.9: Decomposition of cell boundary W_i into its disjoint sets, at the border of region to be surveilled A from [16].

holds true that $v^i_j = v^i_i$ and $n_j = -n_i$, where n_i represents the outward pointing normal vector of W_i. We can finally, by combining the sums and the integrals, reach the final form of the planar control law. Represented in Eq. 5.29

$$\frac{\partial O}{\partial f_i} = \int_{\partial W_i \cap \partial P} n_i q(z_i)\phi(f)df + \sum_{j \neq i} \int_{\partial W_j \cap \partial w_i} v^i_i n_i (q(z_i) - q(z_j))\phi(f)df \tag{5.29}$$

Similarly, for the altitude control law, if we use the same ∂W_i and define $v^i_j(f)$ such that

$$v^i_j(f) \equiv \frac{\partial f}{\partial z_i}, \quad f \in \partial W_j, \quad i, j \in I_n. \tag{5.30}$$

Then the altitude control law will be

$$\begin{aligned} \frac{\partial O}{\partial z_i} &= \int_{\partial W_i \cap \partial P} v^i_i n_i q(z_i)\phi(f)df \\ &+ \sum_{j \neq i} \int_{\partial W_j \cap \partial W_i} v^i_i n_i (q(z_i) - q(z_j))\phi(f)df + \int_{W_i} \frac{\partial q(z_i)}{\partial z_i}\phi(f)df \end{aligned} \tag{5.31}$$

This Eq. 5.31 represent the control law for the altitude, which accordingly to literature [19] and [16] methods, can be simplified into into a final altitude control law form, represented in Eq. 5.32

$$\begin{aligned} \frac{\partial O}{\partial z_i} &= \int_{\partial W_i \cap \partial P} \tan(a) q(z_i)\phi(f)df \\ &+ \sum_{j \neq i} \int_{\partial W_j \cap \partial W_i} v^i_i n_i (q(z_i) - q(z_j))\phi(f)df + q_d(z_i) \int_{W_i} \phi(f)df. \end{aligned} \tag{5.32}$$

Concluding, using drones kynodynamic model defined on Eq. 4.1 and Eq. 4.2, their sensing performance represented in the Eq. 4.3, the coverage criterion present in Eq. 5.21, the partitioning scheme on Eq. 5.20, and what was assumed in chapter 4, the gradient based control laws are the junction of Eq. 5.24 and 5.25 with Eq. 5.29 and 5.32, respectively. Resulting [16] [19]:

$$\begin{aligned} u_{i,f} = \alpha_{i,f}[\int_{\partial W_i \cap \partial P} n_i q(z_i)\phi(f)df \\ + \sum_{j\neq i} \int_{\partial W_j \cap \partial w_i} v_i^i n_i (q(z_i) - q(z_j))\phi(f)df] \end{aligned} \tag{5.33}$$

$$\begin{aligned} u_{i,z} = \alpha_{i,z}[\int_{\partial W_i \cap \partial P} \tan(a) q(z_i) = \phi(f)df \\ + \sum_{j\neq i} \int_{\partial W_j \cap \partial W_i} v_i^i n_i (q(z_i) - q(z_j))\phi(f)df + q_d(z_i)\int_{W_i} \phi(f)df] \end{aligned} \tag{5.34}$$

where criterion presented on Eq. 5.21 is maximized monotonically along with the drone's trajectories, leading always the system to a local optimal solution.

5.2.3.2 Communication

Once as previously saw in Fig. 5.9 the cell of the drone i is affected only by its neighbors drones N_i resulting this way the spatial coordination control law in a distributed control law. In this method approach, the discovery of the neighbors it is not based on the classical Voronoi tessellation or Delaunay triangularizations neighbors search, but in their coordinates and neighbor coordinates $X_j, j \in N_i$. This way, the generation of N_i sets demands drone i to be able to communicate with all the surrounding drones within a sphere centered in the coordinates X_i, which is where the on-board antennas are, and radius r_i^c such that (using sphere geometry)[16],

$$\begin{aligned} r_i^c = \max_i \{2z_i \tan(a), \sqrt{(z_i + z^{min})^2 \tan(a)^2 + (z_i - z^{min})^2}, \\ \sqrt{(z_i + z^{max})^2 \tan(a)^2 + (z_i - z^{max})^2}\} \end{aligned} \tag{5.35}$$

Because all the drones are assumed homogeneous, then the sensors and altitude constraints are the same, resulting in each drone to only broadcast, its state X_i with sufficient signal strength so that all drones within a distance of the radius of r_i^c from drone i are able to receive its message [16].

The neighbors N_i of a drone i are those drones that affect the cell W_i of a drone i and at the same time are those drones whose cells are affected by the drone i. Thus if each drone transmits its state information to its neighbors, then all the drones will be receiving the state information from their neighbors. Thus, in order to generate the set N_i, it is sufficient for a drone i to transmit its state with sufficient power so that it can be received at a radius r_i^c around it and that it is capable of receiving transmissions from other drones [16].

5.2.3.3 Conclusion

From this section is important to highlight that the spatial coordination requires the drones to be controlled through a set of assumptions, that leads the system to look for always the local optimal solution. For this, it is necessary that all the components of the algorithm work together, the spatial control in the planar region and in altitude, the communication between drones and the spatial partition and the network. Also note, that when the altitude of a drone is equal to the maximum height then the control laws will be 0 which means that in this method the drone will be obsolete, decreasing the optimality of the system. However, a solution for this problem will be conducted in next section.

5.2.4 Altitude Stability

Accordingly to the aforementioned control laws if the drone's altitude $z_i = z^{max}$ the drone would be obsolete for the system as its $\phi(f) = 0$. We want to guarantee, in this method, that does not happen because will lead to non optimal coverage. For that, we examine the drones altitude z_i assuming all the points in area A have $\phi(f) = 1, \forall f \in A$, and show that it always remains in the interval of $z_i \in [z^{min}, z^{max}]$. Thus, we need to study the behavior of $\dot{z}_i$ over the time and system state [16].

For a team of drones, each drone will move towards an altitude that is an equilibrium point of the system, that is, accordingly to [16]:

$$\dot{z}_i = u_{i,z} \quad , u_{i,z} \in \Re \tag{5.36}$$

where,

$$u_{i,z} = \int_{\partial W_i \cap \partial P} \tan(a) q(z_i) df + q_d(z_i) \int_{W_i} df + \sum_{j \neq i} \int_{\partial W_j \cap \partial W_i} \tan(a)(q(z_i) - q(z_j)) df. \tag{5.37}$$

This altitude that confers the system the equilibrium point is defined as z_i^{stb}. The stable altitude is not common among all drones as it depends on one's neighbors N_i and is not constant over time since the neighbors change over time.

In order to understand the behavior of altitude stability control we need to see the behavior of each drone alone. For that its useful to define an optimal altitude z^{opt} as the altitude that a node would reach if: firstly, had no neighbors $N_i = \emptyset$ and secondly its whole cell was inside the region of interest $A \cap W_i = W_i$. When the aforementioned requirements are met it holds true that $W_i = C_i^s$. This optimal altitude is the stable equilibrium point of the system and its defined by [16]:

$$\dot{z}_i = u_{i,z}^{opt} \quad , u_{i,z}^{opt} \in \Re \tag{5.38}$$

where,

$$u_{i,z}^{opt} = \int_{\partial C_i^s} \tan(a) q(z_i) df + q_d(z_i) \int_{C_i^s} df + 2\pi \tan(a)^2 z_i q(z_i) + \pi \tan(a) z_i^2 q_d(z_i). \tag{5.39}$$

This altitude is constant and depends solely on the network minimum and maximum height. As the planar control law exposed in last section has a behavior close to the DSSA presented in chapter 3, where if $A = \Re^2$. The repulsion between nodes occurs until the network reach a state in which no node will have neighbors and all drone will be at z^{opt}. In that state, the coverage quality function's criterion in Eq. 5.8 will be at his maximum possible value. For that network configuration and coverage quality function, the network will be at his best solution (globally).

In the case, when A is a convex compact subset of $\Re^2$ it is possible for the network to reach a state where all the nodes are at the optimal height, only if a determined number of drones can be packed inside A. This state will be globally optimal. If that is not the case, the nodes will converge at some altitude rather than z^{opt} and in general different among drones. It should be noted although that despite the drones do not reach the optimal height, the network configuration is still a local optimal solution.

Analyzing the system stability properties defined in [16] lead us to the Fig. 5.10 where $u_{i,z}^{opt}(z_i)$ is shown in blue, in green is the integral over the drone i cell, and in red, the integral over the boundary of the drone i cell.

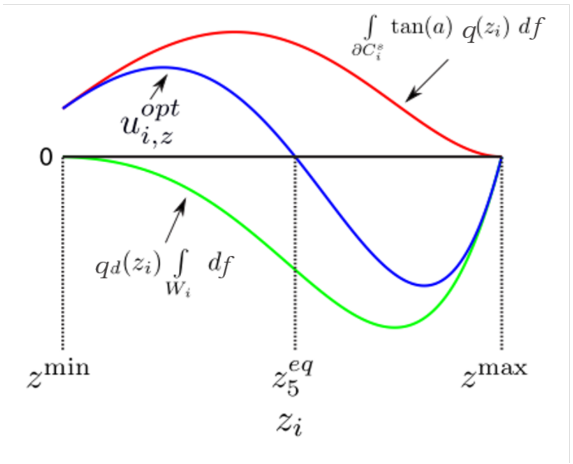


Figure 5.10: Plot of the system optimal altitude for drone i and in terms over W_i and ∂W_i with respect to drone actual height from [16].

From there we extract that z^{max} is unstable because a small negative disturbance dz will result in the drone to lower his altitude and depart from the maximum height of network. Similarly, the equilibrium point z_5^{eq} is stable because a small negative disturbance dz will result in the drone to an higher altitude and closer to z_5^{eq}. On the other hand, a small positive disturbance will dz will lead the drone to a lower altitude and again, closer to z_5^{eq}. In conclusion, when a node has no neighbors and its whole cell is inside the area to be surveilled, the only stable point is the the optimal point [16].

For multi drones altitude stability, we will assume the behavior seen previously for only one drone but this time with neighbors. Also, the system we examine for this situation is the one that derives from the Eq. 5.38 and 5.39. The integrals over the drones cells boundaries are non-negative whereas the integrals over the drones cells are non-positive. Denote that, the integrals over the boundary of a cell of a drone with neighbors will always be smaller than the same integral for a situation without neighbors because the neighbors will either remove some arcs of the cell from the integral or reduce their influence due to the term $q(z_i) - q(z_j)$. Similarly, the absolute vale of the integral over a drone cell with neighbors will not be bigger than the same for drone cells without neighbors because the area of the cell bellowing to the sensed region has been assigned to neighbors with higher coverage image quality. Thus we conclude that z_i^{stb} will attain its minimum value when the integrals over the cell boundary are zero and its maximum value when integrals over the cells are zero.

When the integrals over the cell boundaries are both zero, the altitude control law has a negative value. This will lead to a reduction of the drone's altitude and in time the node will reach the $z_i^{stb} = z^{min}$. Once the drone reaches z^{min} its altitude control law will be zero until the integral over the boundary of the cell stops being zero. When the integral over the drone cell is zero, the altitude control law, has a positive value. Therefore, the drone's altitude will be increased and in time the drone will reach $z_i^{stb} = z^{max}$ and has referred before, the node will be immobilized from this time onwards. When the integral over the drone cell and at least one of the integrals of the drones cell's boundaries are non-zero then the system will maintain its stability just like the example of the drone that had no neighbors [16].

Due to the drones initial positions, as in this method, we assume that the drones are already inside the area A and we don't have to worry about taking them there, we cannot be sure that their initial coordinates wont be problematic for the system. In order to deny this problem, we should be careful not to put drones initial positions such that one drone sensed range is completely inside another drone cell, the drones get no responsibility or simply two drones are so infinitesimal close to the other that the system only counts the one with better quality coverage [16].

5.2.5 Algorithm

In algorithmic view, this method of communicative area coverage has many levels of implementation accordingly to [19] and [16]. On a first level, we need to create the full system layout so we can insert the many different algorithms such as the control laws, the altitude stability, space partitioning scheme and the coverage quality function. In order to implement the aforementioned method we need to approximate the sensed region, which was assumed to be a circle, into a polygon using the equations presented in [16] and being represented by $\sim C_i^s$. In the same way, each cell boundary is approximated by a polygon $\sim W_i$. The border of the area to be surveilled is also approximated by a polygon $\sim A$.

The algorithm 2 presents the space partitioning strategy and the construction of the drone i cell. In algorithm 3 the calculations for the control laws are made. In order to calculate these control laws, several line integrals have to be calculated numerically as well as one double integral. The line integrals are calculated as sums. The double integral is just the area of the corresponding drone cell and can thus be calculated simply as the area of the polygonal approximation.

Algorithm 2: Partitioning - Construction of cell W_i^p of node i [16]

Input: $X_i, X_j \forall j \in N_i$
Output: $\sim W_i$
$\sim W_i \leftarrow \sim C_i^s$
foreach $j \in N_i$ **do**
  if $q(z_i) \leq q(z_j)$ **then**
    $\sim W_i \leftarrow \sim W_i \setminus \sim C_i^s$
  end
  else if $q(z_i) = q(z_j)$ **then**
    $H_{ji} = \{f \in A : ||f - f_j|| \leq ||f - f_i||\}$
    $\sim W_i \leftarrow \sim W_i \setminus \sim H_{ji}$
end
$\sim W_i \leftarrow \sim W_i \cap \sim A$

Algorithm 3: Control - Calculation of control laws $u_{i,f}$, $u_{i,z}$ [16]

Input: $X_i, X_j \forall j \in N_i$
Output: $u_{i,f}, u_{i,z}$
while *Transmit* X_i *and receive transmissions from other agents* **do**
  $u_{i,f} \leftarrow 0, u_{i,z} \leftarrow 0$
  execute algorithm 2 - PARTITIONING
  foreach *edge* $E_{i,k}$ *of* $\sim W_i$ *with vertices* e_{k1} *and* e_{k2} **do**
    if e_{k1}, $e_{k2} \notin \sim A$ **then**
      if $e_{k1} \in \sim C_i^s \vee e_{k2} \in \sim C_i^s$ **then**
        compute length of $E_{i,k}$ as $dl = ||e_{k1} - e_{k2}||$
        compute outwards unit normal vector n_{ik} of edge $E_{i,k}$
        if $\exists j \in N_i : e_{k1}, e_{k2} \in \sim W_j$ **then**
          $u_{i,f} \leftarrow u_{i,f} + n_{ik}(q(z_i) - q(z_j))dl$
          $u_{i,z} \leftarrow u_{i,z} + \tan(a)(q(z_i) - q(z_j))dl$
        else
          $u_{i,f} \leftarrow u_{i,f} + n_{ik} q(z_i) dl$
          $u_{i,z} \leftarrow u_{i,z} + \tan(a) q(z_i) dl$
        end
      end
    end
  end
end

The system overall algorithm is exposed in algorithm 4 where its shown how the different algorithms are sequenced in order to make this method work.

Algorithm 4: Overall communicative area coverage method simulation

```
Input: Drones Initial Positions
Data: Angle of cameras, Control laws gain,
camera Footprint, Number of algorithm iterations
Output: Drones Final Positions
Result: Optimal positioning for maximizing the area coverage and coverage image quality
Initialize region
Initialize network
Initialize simulation
while simulation is running do
    for each drone do
        Create covered region
        Calculate coverage quality
    end
    for each drone do
        Calculate its assigned region to surveil
    end
    Calculate the value of the optimization objective O
    Plot the multi-drones system state
    for each drone do
        Calculate the control inputs
        Move accordingly to the control law inputs
    end
end
```

5.3 Conclusion

In this chapter 5 we explained the two methods essentially developed by [18] [17] for the Non-communicative offline area coverage, and by [16] and [19] for the communicative area coverage. These methods propose to solve the problem presented in chapter 4. The first problem, presented in section 4.1, started by showing the area decomposition technique based on [17] literature. Then, coverage problem was modeled as a graph and then into a TSP problem. In order to solve it, variables and constants were defined, followed by the objective function (which was a min-max function situation) and last the constraints of the system. The solution of the TSP problem which is a NP-hard with a global optimal solution and not local optimal solution starts by running the MILP iteratively, removing the nodes of the best solution and running the TSP problem again until the base is the only node available. This way, the system gives us the optimal global solution. Conclude also, that for this problem, the FWUAV were the more appropriated and in order to get the best share of the workload in the drones team was needed to know what the number of operators is or, the setup time and flight autonomy of the drones, assuming that they are homogeneous. In conclusion, we presented an algorithm for the method overall for further implementation of the non-communicative area coverage problem.

The second problem, presented in section 4.2, its about covering an area with focus on communication, space partitioning, coverage quality, Control Laws and Stability control. To solve the problem imposed, the coverage quality function was defined along with all data known in advance. With the objective of maximizing the coverage quality and the area covered at the same time, an algorithm for area partitioning was proposed based on the sensed area by the drones team. Despite the holes and gaps in the coverage, this coverage gives optimum real-time ground images of the area surveilled. This is done by managing collaboratively the sensed region cells among the drones. The best the image, the higher is the objective function. To understand best how the drones interact, in this method was presented the control laws of each drone and the relation of his behavior with the others. In order for each drone to know what the others are doing a communication network is established on *peer-to-peer* style, where one drone in a given space can communicate with their surrounding neighbors and so on. Creating this way a distributed collaborative space partition and coverage algorithm. For last, altitude stability was examined firstly with a drone inside the area of interest with no neighbors and secondly with a drone team with neighbors. Relations were made, proving that the system always moves the drones into stable altitudes that represent an increase in system performance. For last, some problem that could arise from the algorithm were delimited by carefully position the drones on initial states. A couple of algorithms were then exposed in order to understand how to implement and simulate the method proposed in this chapter for communicative area coverage.

Chapter 6

Simulations and Analysis

This Chapter 6 illustrate through computer simulations the area coverage problems proposed in Chapter 4 using the methodologies exposed in Chapter 5. Simulations of the two methodologies take into account different scenarios, where we show the optimized solution accordingly to the methods described in Chapter 5 .

Firstly we show the simulation results for the non-communicative area coverage algorithm [18], exploring the constraint's effect on the global performance of the method, followed by the analysis of the results obtained. Similarly to non-communicative area algorithm, we explore the different situations to put in test the communicative area coverage method [16] [19]. In this last one, we will check different cases and analyze how the system works and how its supposed to be working. For last, we will make a brief comparison between the two algorithms accordingly to the simulations and respectively results.

All simulations on this work were executed in MATLAB R2015b running in an Intel(R) Core(TM) i5-3570k CPU @ 3.40GHz 3.80GHz computer with 8 GB of RAM operating under Windows 10.1 (64bits).

6.1 Non-communicative offline area coverage

This section pretends to illustrate the simulations of the proposed methodology using a series of simulations. Each simulation will set into a different aspect of the system where, for the same system specifications, the performance is evaluated with and without certain constraints defined. Finally we will evaluate the impact of different constants assumptions on the overall performance, and for different drone team's model.

6.1.1 System specifications

To compare the algorithm results, we considered that all drones are homogeneous FWUAV or homogeneous rotary-wing UAV.

The choice of the vehicles specifications is based on LSTS and we considered the X8 Skywalker and the Mariner. Accordingly to LSTS [15] X8 Skywalker is a commercial of the shelf delta wing drone perfect to be used in surveillance missions. In Fig. 6.1 an X8 drone can be seen and in Table. 6.1 its technical features are presented. Similarly, the chosen rotary-wing UAV is the Mariner which is a quadrocopter also good on area coverage missions. An example of a Mariner can be seen in Fig. 6.2 and its specifications are presented in Table 6.2.

Figure 6.1: X8 Skywalker from LSTS [15].

Table 6.1: X8 Skywalker's Specifications from LSTS [15]

X8 Skywalker	
Wingspan	212 cm
Length	60 cm
Weight	Starting at 1.5 up to 3.5 Kg
Endurance	Up to 60 min
Maximum Altitude	600 m
Speed	85 Km/h
Wind tolerance	Mean 14 Kts l Max 18 Kts
Camera	HD 720p on-board recording and streaming
Communication	Wi-Fi 2.4 Ghz

Figure 6.2: Mariner from LSTS [15].

Table 6.2: Mariner's Specifications from LSTS [15]

Mariner	
Length	55 cm
Weight	Starting at 3.5 up to 4 Kg
Endurance	Up to 20 min
Maximum Altitude	1000 m
Speed	58 km/h
Wind tolerance	Max 19 Kts
Camera	GoPro3 HD 1080p recording on-board and analogical live streaming
Communication	Wi-Fi 2.4 GHz and radio RC 2.4GHz

Choosing two different type of drones will allow us to get some insight of how the design and specifications impact our area coverage method, when using two different teams of different drones. Therefore, allowing to conclude which is the best design for this mission.

Knowing this, and using the actual laws applied over drone's hobbyist. We defined two different system specifications for the simulations, one considering a team of FWUAV and other a team of quadrocopters as shown in Table 6.3.

Table 6.3: System's Default specifications

	FWUAV's Team	Quadrocopter's Team
Number of Operators	1	1
Number of Drones	3	3
Drone's Setup Time	2 min	1 min
Drone's Flight Time	60 min	20 min
Drone's Flight Speed	85 km/h	58 km/h
Flight Altitude	300 m	300 m
Camera's Resolution	15 MP	15 MP
FOV (mm [18])	35 mm	35 mm
Image Overlapping Rate	30 %	30%

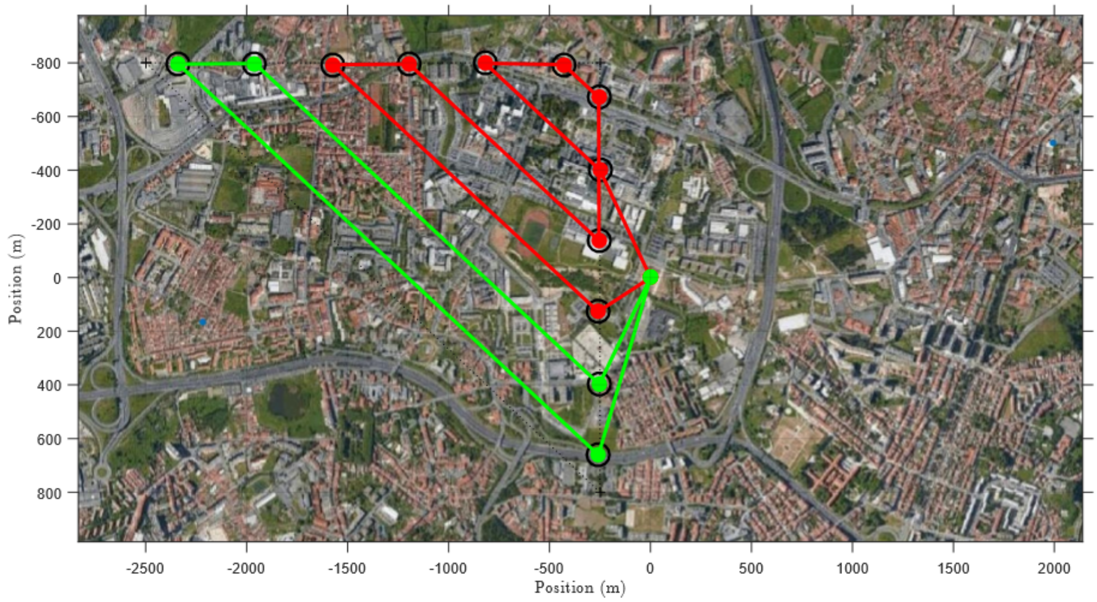
6.1.2 Simulation and Results

The simulations made take into consideration, in first place, the constraint made on the coverage problem and their impact on global solution. In second place, we examine the behavior of the system when some constants are changed.

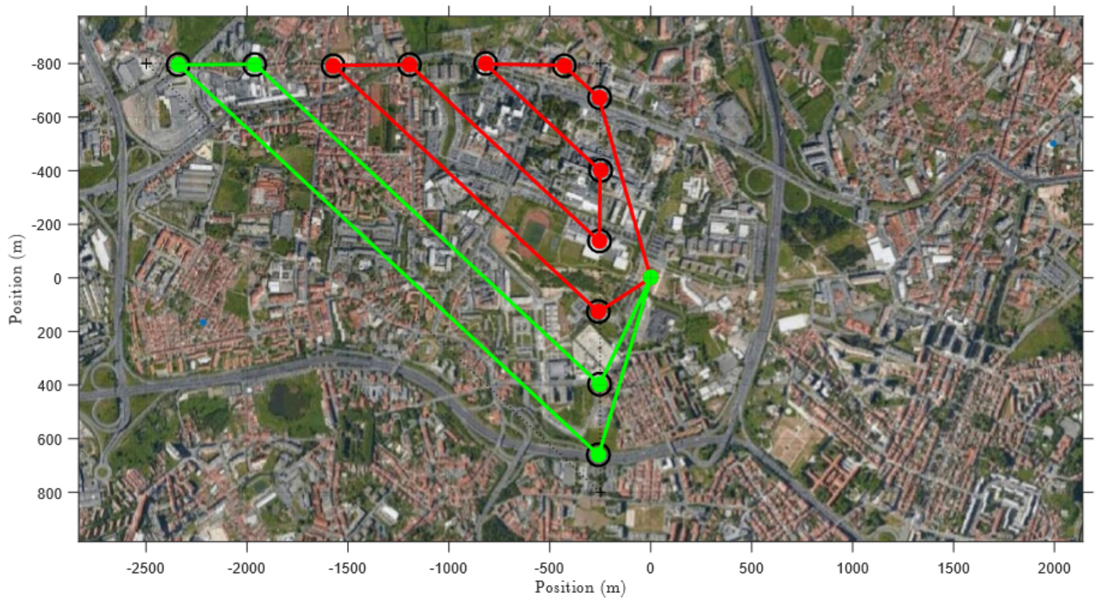
Simulations were done using MATLAB for the problem formulation and data treatment, YALMIP Toolbox for the optimization problem front-end and GUROBI Solver for the optimization problem back-end, and plotted on Google map. Some simulations run at Asprela's Campus and others at Porto's downtown. Until opposite, the simulations specifications are the ones defined in Table 6.3.

6.1.2.1 Algorithm Constraints

In order to prove the algorithm's methodology presented in Section 5.1 throughout the Eq. 5.6 to Eq 5.17 we put the constraint to test and see the results obtained when they are not considered and when they are considered. Note that in this case the drone type is not relevant. Just the setup time, the number of operators and the drones. Flight speed was chosen as a average value between the aforementioned drones velocities (60 km/h).



(a) The mission without iteration.



(b) The mission with iteration process.

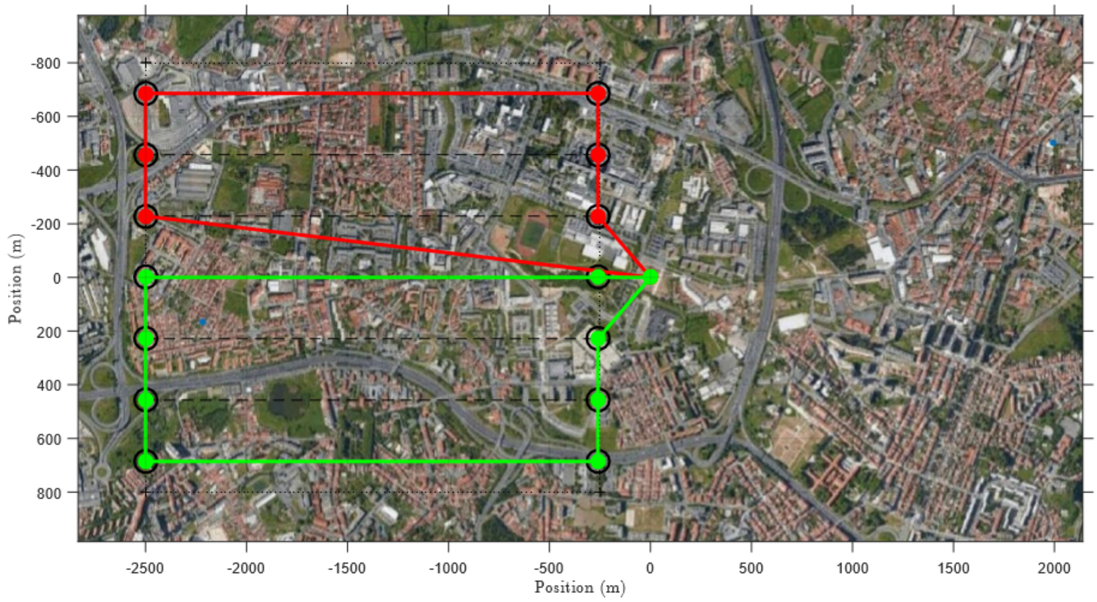
Figure 6.3: Impact of the iterative method on Eq. 5.6 on the overall performance of the system.

On first test (Fig. 6.3), we explore the Objective function 5.6 and show that running the optimization problem just once will lead to the minimization of the maximum result, which mean that only the longest route is minimized. This will cause the system to find optimal solution for the longest route, but not for all routes. In order to do that, we iteratively remove the best route until all routes are optimal leading to the optimal solution. We consider in this simulation that two drones with no setup time are sent to cover a triangular-shaped area. From Fig. 6.3a we see that when the heuristic iterative method is not applied, only the longest path is minimized. Thus, the drone that makes the route plotted in green (longest path) is optimal, corresponding to a mission of 6 minutes and 10 seconds. In other hand the drone that makes the route plotted in red, has unnecessary cycles and its not optimized corresponding to a mission of 5 minutes and 43 seconds. In Fig. 6.3b the problem is solved. Once the longest path was already optimized, running the second iteration will make the biggest second path to be optimized, in this case, the red one, avoiding unnecessary cycles which corresponds to a mission of 5 minutes and 41 seconds. In this case, the optimization was low in term of percentage, however in other situations may have great impact.

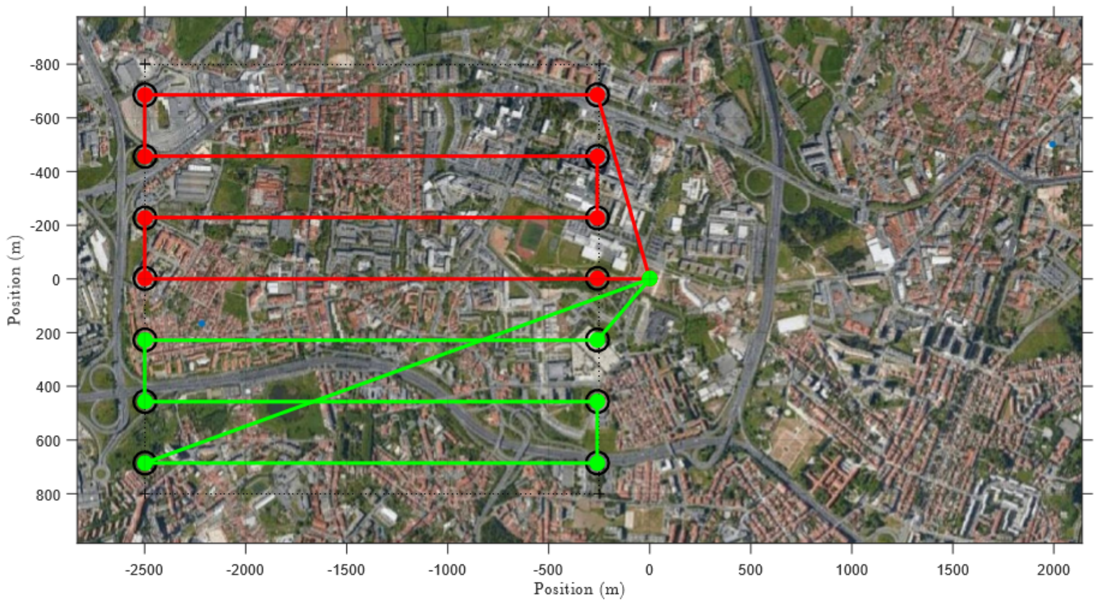
On our second test (Fig. 6.4), the constraint in Eq. 5.13 is explored to prove that the constraint impose the system to always be a coverage problem and not one TSP problem, despite treated as one in this method, by saying that a drone that visit a graph node in a row must visit the opposite node of that same row. To see the impact of this constraint on the area coverage method we assumed the same situation in the previous test, but this time, the area to be surveilled is a simple rectangle. From Fig. 6.4a we can see that when we obliterate the constraint in Eq. 5.13 the drones travels all nodes in minimum time but its not taking into account that the area need to be covered, and for that they need to follow the rows. This application would be interesting for area border control, however we look for total area coverage on this method. In Fig. 6.4a we see that the constraint examined makes the drone to follow the rows.

On our third test (Fig. 6.5), we draw our attention to the optional constraints exposed in Eq. 5.14 and 5.15 as they remove the possibility of drones to follow paths where they would cross diagonally the region to be covered, except in case of returning or leaving the base. Although this may increase the time to cover the region and the energy consumption as shown in [23], through the increased curves, they also create very sharp angles on curves which is not desired if we want FWUAV to operate on this area coverage method. Moreover, the fact that the drones do not cross the region diagonally reduce the area coverage redundancy and possibility of collision with other drones. For this simulation we used two drones with no setup time covering a region similar to the one in second test. In Fig.6.5a we see that allowing drones to cross the rows diagonally minimize mission total time, which corresponds to 7 minutes and 58 seconds for both drones routes. Applying the constraints on the system method, result in Fig. 6.5b solution, where the paths do not cross diagonally and the optimal solution corresponds to 8 minutes and 4 seconds for each drone route. Note that, these constraints reduce the complexity of the optimization problem once they eliminate from the modeled graph the diagonals between the nodes except base's node.

In next tests, we prove that one of the big contribution of the methodology proposed and shown in Chapter 5 has a major contribution on practical tests which normally are not considered in simulations. In this test, the constraint defined on Eq. 5.8 for the extra time taken for another



(a) The method as a TSP problem.

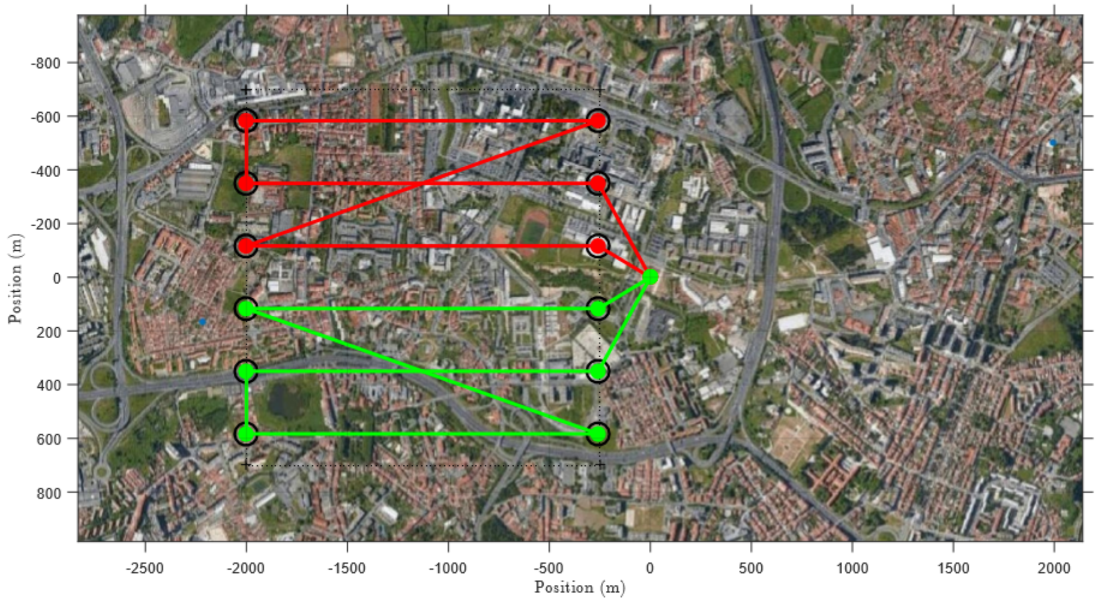


(b) The method as an area coverage problem.

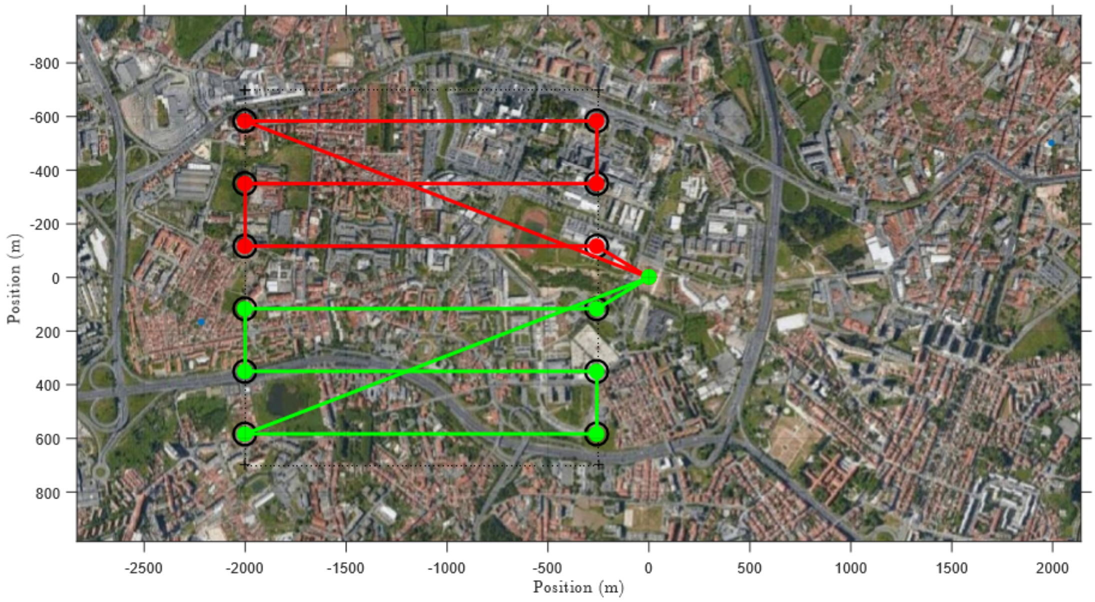
Figure 6.4: Impact of the iterative method on Eq. 5.13 on the overall performance of the system.

drone to set flight, is very helpful to realize that in a real scenario, where the number of operators is less than the number of drones, the system still be optimized and with low margins of delay.

In Fig. 6.6 we assume that for the same area and drone specifications, we have a team of four drones, and one operator, the difference is on the setup time, in first case, presented in Fig. 6.6a theres the result from simulating the area coverage method considering the drones have no set up time, for example can take off autonomously. We see in this case that the time taken by four drones at a average velocity of 60 km/h to cover Porto's downtown with depot on "Serra do Pilar"



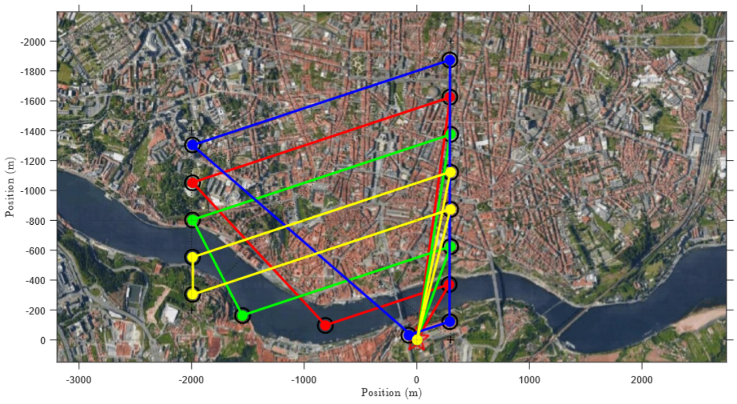
(a) The method with diagonal paths between row's nodes.



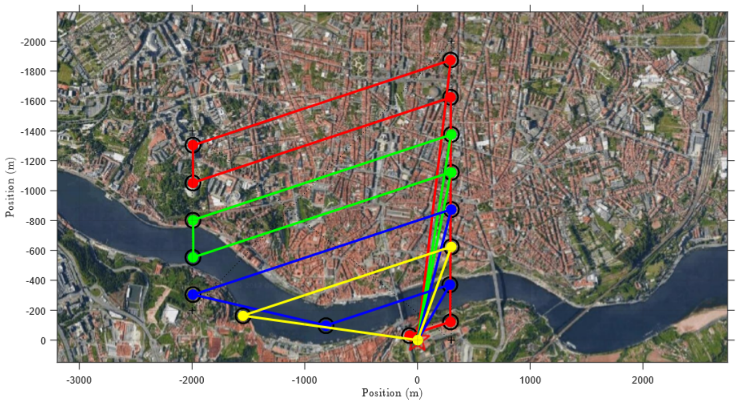
(b) The method without diagonal paths in exception of base's node.

Figure 6.5: Impact of the constraints on Eq. 5.14 and 5.15 on the overall performance of the system.

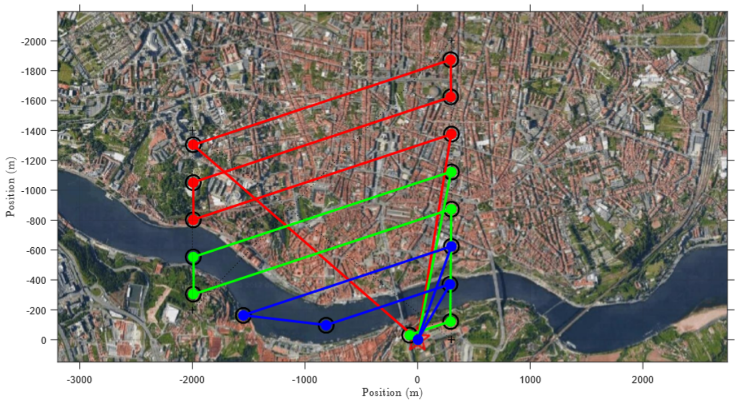
was 7 minutes and 8 seconds for the red route, 7 minutes and 8 seconds for the green route, 6 minutes and 56 seconds for the blue route and lastly 7 minutes and 3 seconds for the yellow route taking a total time of 28 minutes and 15 seconds. When considering the same situation, but taking into account the setup time (considered one minute and a half based on average time considered from Table 6.3) of one minute and a half, the results, present in Fig. 6.6b, correspond for red, green, blue and yellow to, 8 minutes and 48 seconds, 7 minutes and 32 seconds, 6 minutes and 5 seconds and 4 minutes and 9 seconds, respectively taking a total time of 28 minutes and 4 seconds.



(a) The coverage of Porto's Downtown considering a team of four drones and no setup time.



(b) The coverage of Porto's Downtown considering a team of four drones and a setup time of 1 minute and a half.



(c) The coverage of Porto's Downtown considering a team of four drones and a setup time of 3 minutes.

Figure 6.6: Impact of the setup time on number of drones deployed and resulting minimization of objective function in Eq. 5.6

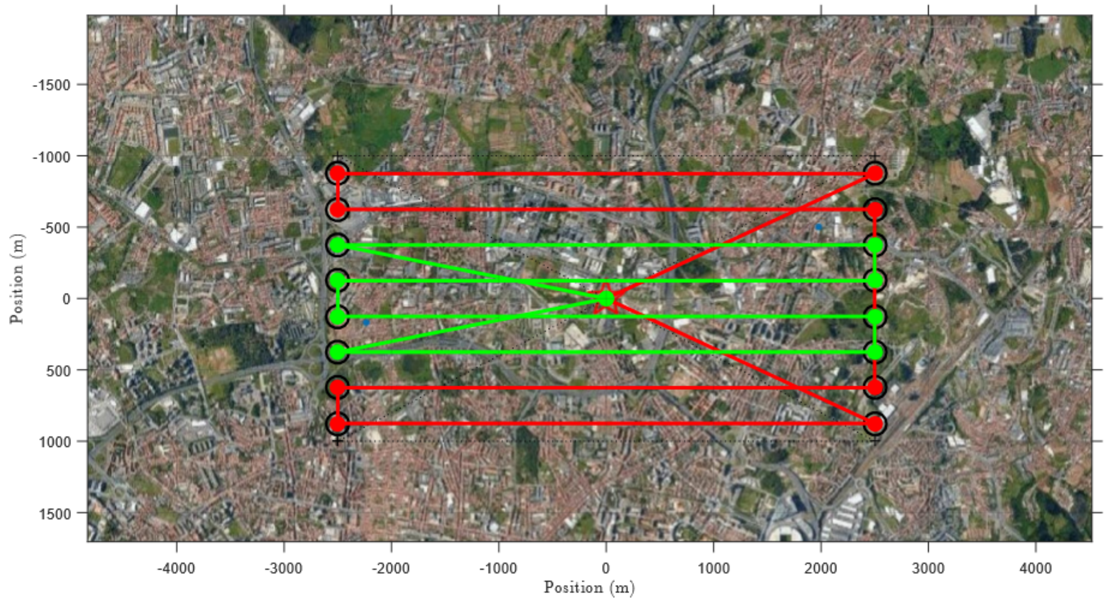
Despite the workload is more equally distributed on first scenario considered, the second scenario distributes realistically the coverage mission and has better performance. Moreover, the algorithm implemented can have even better performance by making use of only the strictly necessary drones if the drones setup times are to high and the use of one extra drone is not worth, despite the work overload. In the same example scenario, as mentioned before, is presented in Fig. 6.6c, a situation where the setup time are higher, in this case, 3 minutes. The effect of constraints in Eq. 5.16 and 5.17 will lead the solution of the problem not to use all the available drones, thus reducing the time taken to perform the mission having a better overall performance compared to situations where all the drones had lower setup times, like in test in Fig. 6.6b. This last simulation, corresponds to 11 min and 21 seconds for the red drone route, 7 min and 69 seconds for the green drone route and 4 minutes and 56 seconds to the last one drone route resulting in a total of 26 minutes and 36 seconds. Despite there are four drones available the method concluded that it was worthy in time and cost to only use three.

6.1.2.2 Algorithms Drawbacks

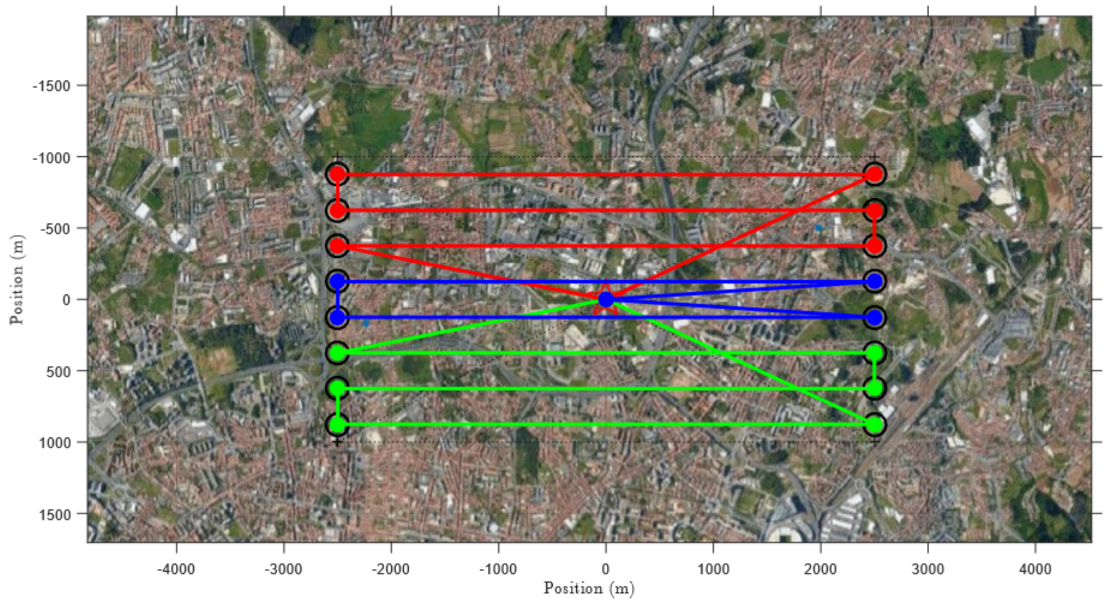
The strategy previously illustrated also has its drawbacks. One of those drawbacks is the loss coming from a drone to perform a route so long that has to deploy a new one to finish is intended path (if it had enough endurance). Once in this method, from [18], its assumed that a drone cannot recharge in any way and that he only enter on a row if has enough battery to finish. Other miscalculations on this method are the non-employment made over the drone flight time departure from base and on return. This behavior can be seen in Eq. 5.9 which constraints the system solution to look for a solution that complement the drone not to go out of battery. However this situation only happens when the deployment setup time is to high making the deployment of extra drones not optimal and when the area is too big for the used drones endurance. Sometimes it even makes the problem infeasible.

In simulation presented in Fig. 6.7 we explore this issue by defining a big area to cover and, in first situation, considering three drones with moderate endurance (30 min) and very high setup time (40 min, for instance); and secondly three drones with shorter endurance (25 min), but same setup time. Thus, for this first simulation a team of three drones with a flight speed of 60 km/h, only one operator and a big area, ($5000x2000m^2$ for instance). Results, for the first scenario, as seen in Fig.6.7a, is to deploy only two drone that covers all the area by itself, corresponding to 27 minutes and 2 seconds for the red route and 25 minutes and 48 seconds for the green route resulting in a mission total time of 1 hour 45 minutes and 48 seconds. This way, if we consider a lower flight time, not enough for the two drone cover all the area, that will force the operator to deploy the third drone resulting in an huge optimization loss. Thus, the third drone will fly a much shorter path as seen in Fig. 6.7b, where the mission time corresponds to 20 minutes and 40 seconds for the red one, 20 minutes and 40 seconds and 15 minutes and 15 seconds for the last resulting in a total time of 2 hours 15 minutes and 15 seconds. As we see, the lack of endurance can be a big bottleneck in this method optimization.

Other downsize of this algorithm is that due to its non-communicative nature, if we deploy a team of drones to cover an area and one of them has a malfunction or an accident due to external sources, the area will be uncovered and there is no way to contour or compensate that situation once the paths are generated offline with the base station and while in covering maneuver they do



(a) Coverage for a team of 3 drones with setup time of 40 minutes and endurance for 30 minutes

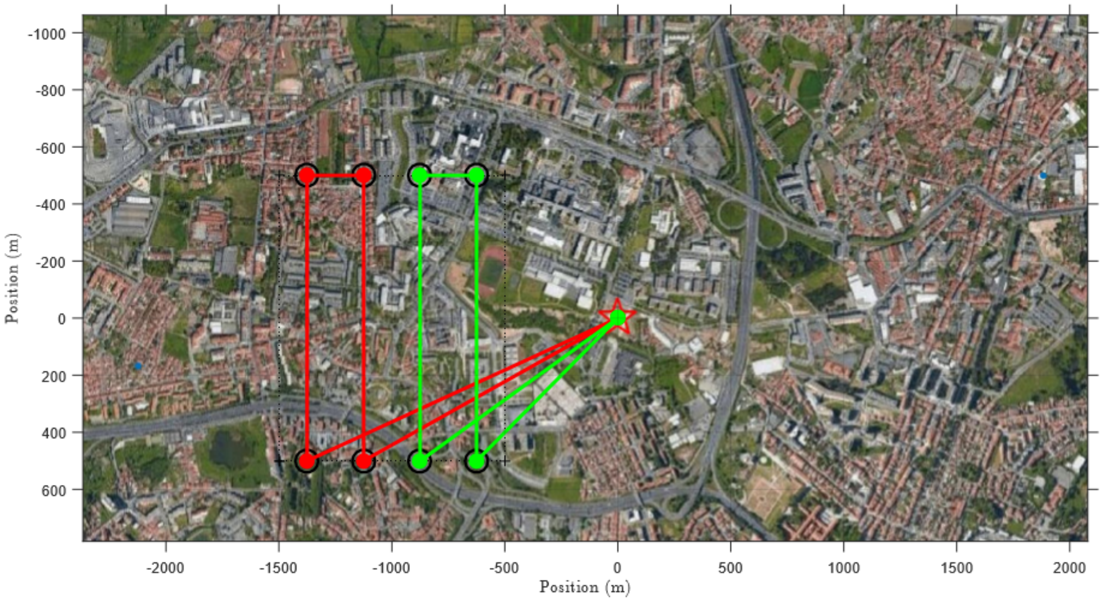


(b) Coverage's bottleneck for a team of 3 drones with setup time of 40 min and endurance for 25 minutes.

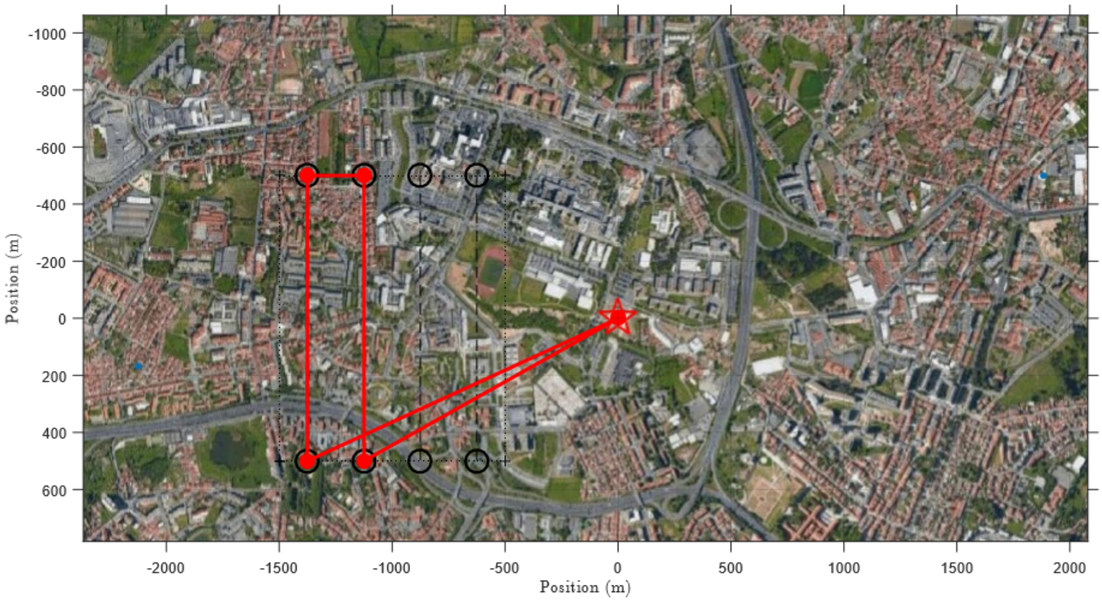
Figure 6.7: Drones lack of endurance when setup times are big, causing big loss in optimization results.

not communicate with other drones. This way, an example is presented in Fig. 6.8, assuming a team of two drones and where 6.8a shows the intended paths to be followed by the drones team and, in 6.8b the actual covered region if the drone represented by the green route has a malfunction.

Last drawback examined is the fact that in this methodology, the non convex areas are covered as if they were convex, this is, the algorithm creates a convex hull around the non-convex polygon. Which mean that just like in Fig. 6.9, we loose to much time to cover areas of no interest, and



(a) Planned routes.



(b) Executed routes.

Figure 6.8: Illustrative example of area coverage percentage lost with a drone malfunction.

reduce the effectiveness of the surveil method.

6.1.2.3 Algorithms Constants

Other characteristic of this area coverage methodology is that the higher the flight altitude the lower are the numbers of rows to be followed once each row surveil a wider path. So it is expectable that for higher altitudes, covering the same area, require less time. As shown in Fig. 6.10 we have the number of rows to a given area, when altitude is 100 meters (Fig. 6.10a) and when 200 meters (Fig. 6.10b). Assuming that in this simulation only one drone is used with no setup

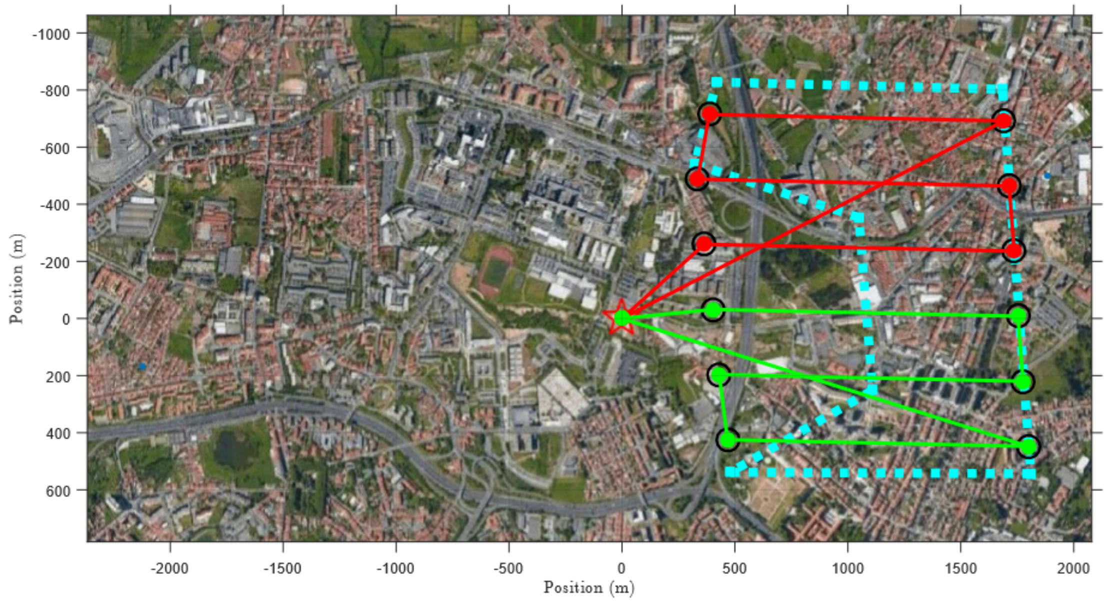


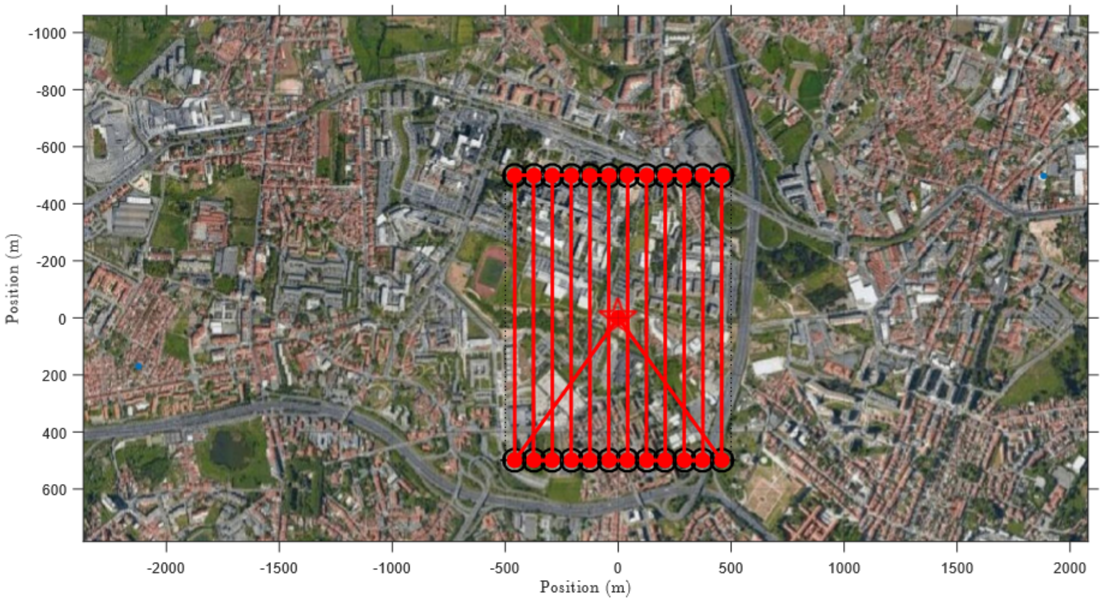
Figure 6.9: Result obtained from covering an non-convex polygon area (Border in Cyan shows the area to be covered).

time. We obtain that, for the simulation in the left side 6.10a the mission time is 14 minutes and 16 seconds and, for the simulation in the right side 6.10b the mission time is 8 min and 8 seconds. Showing a relation of almost half of the time. The number of rows is has expected reduced from 12 to 6.

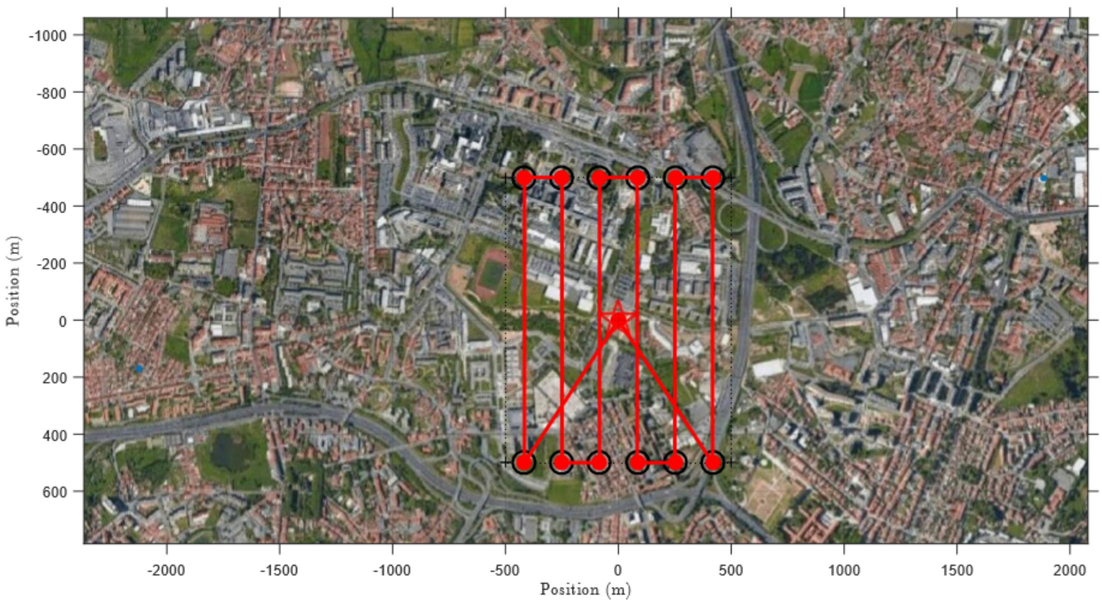
Similarly, the image overlap also define the wide of the covered region as shown in Chapter 5 the number of rows its directly dependent on wide of the covered region by the drone. Which is, posteriorly dependent on camera specs, flight altitude and image overlapping like previously proved by the Eq. 5.2 and 5.3. The expected result is the same as the one in Fig. 6.10 avoiding this way another result plot.

For last, it is expected that the flight speed actuates on how much area can a team of drone cover, if the speed is high then they can cover larger areas for the same endurance. On the other hand the same principle is applicable. In Fig 6.11 two different simulations are exposed, on the first, the flight speed is 30 km/h, on the other one speed is 60 km/h. The results are visible in Figs. 6.11a and 6.11b. The time of the missions is for the first case, 12 minutes and for the second, as expected, 6 minutes.

However this will be not the case in a real implementation because there are external factors like wind's speed and direction,vehicle's acceleration and deceleration on curves to maintain the control when curving, among other factors that would have to be considered.



(a) Number of rows for a certain area at a flight altitude of 100 meters.

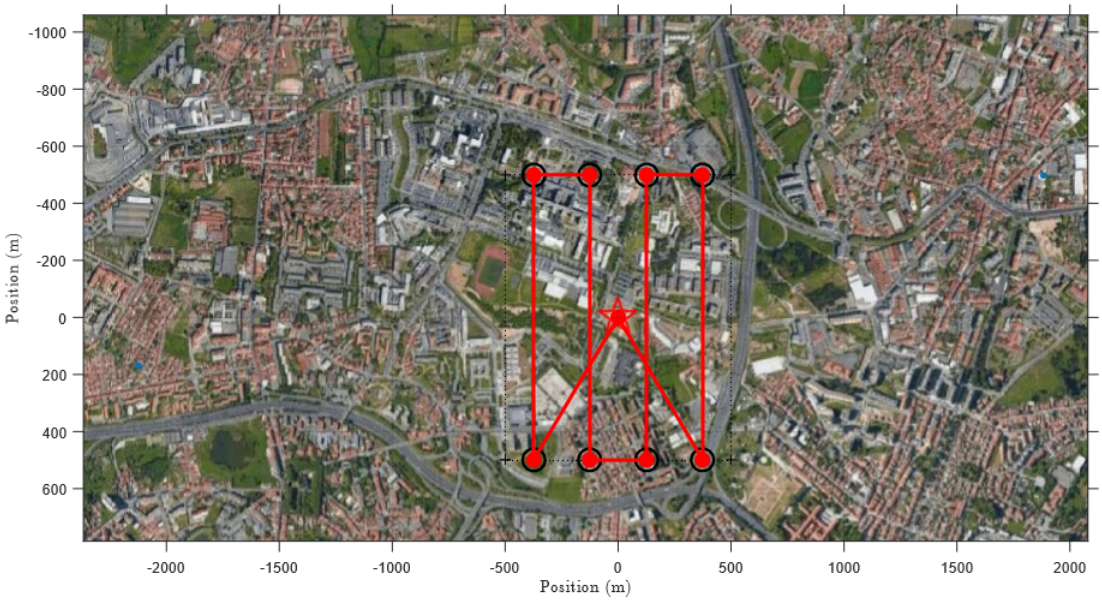


(b) Number of rows for a certain area at a flight altitude of 200 meters.

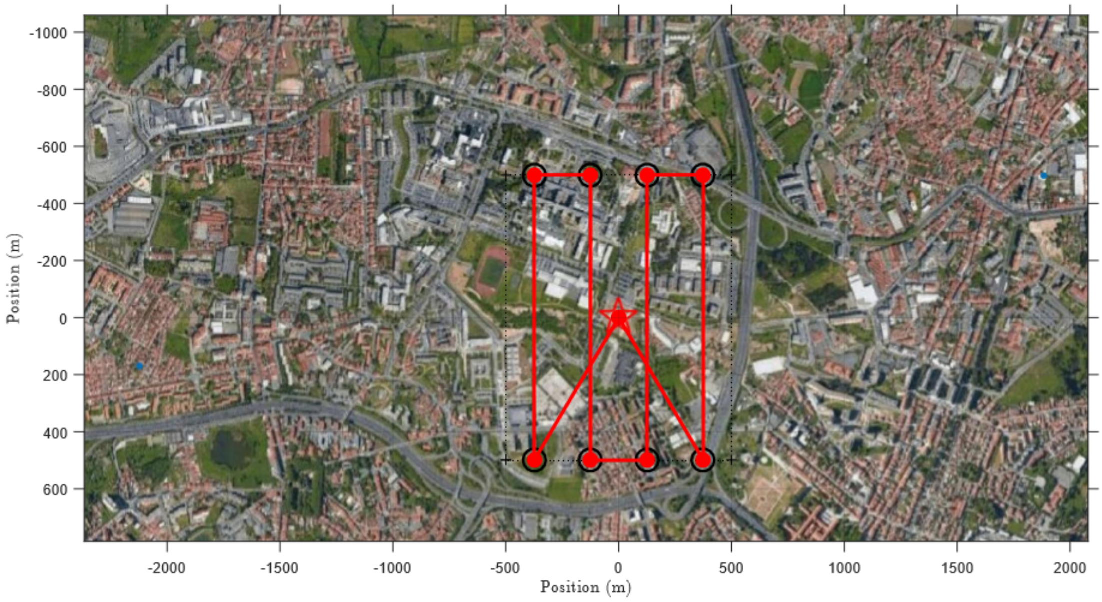
Figure 6.10: Illustrative example of area coverage flight altitude on the number of necessary rows.

6.1.2.4 Drone design results

Inspired on the two different type of drones assumed, namely, the FWUAV X8 Skywalker (Fig. 6.1), and the quadrocopter Mariner (Fig. 6.2) from LSTS [15]. We try to conclude which kind of drone typology best adapts for this area coverage method. Knowing in advance that FWUAV struggle on the curves and sharper turns, but having generally better flights speeds and endurance times, it is expected for them to perform better than rotary-wing UAV, which despite have more flexibility on turns, consume more energy to stay on the air, thus having lower endurance and



(a) Coverage with one drone with a flight speed of 30km/h.



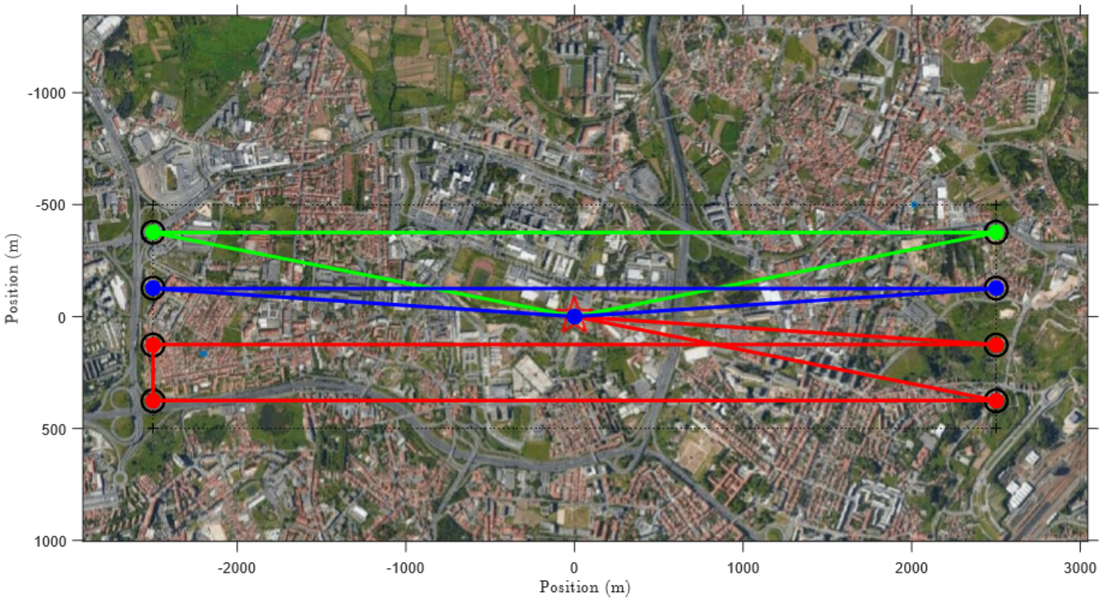
(b) Coverage with one drone with a flight speed of 60km/h.

Figure 6.11: Illustrative example of area coverage flight Speed.

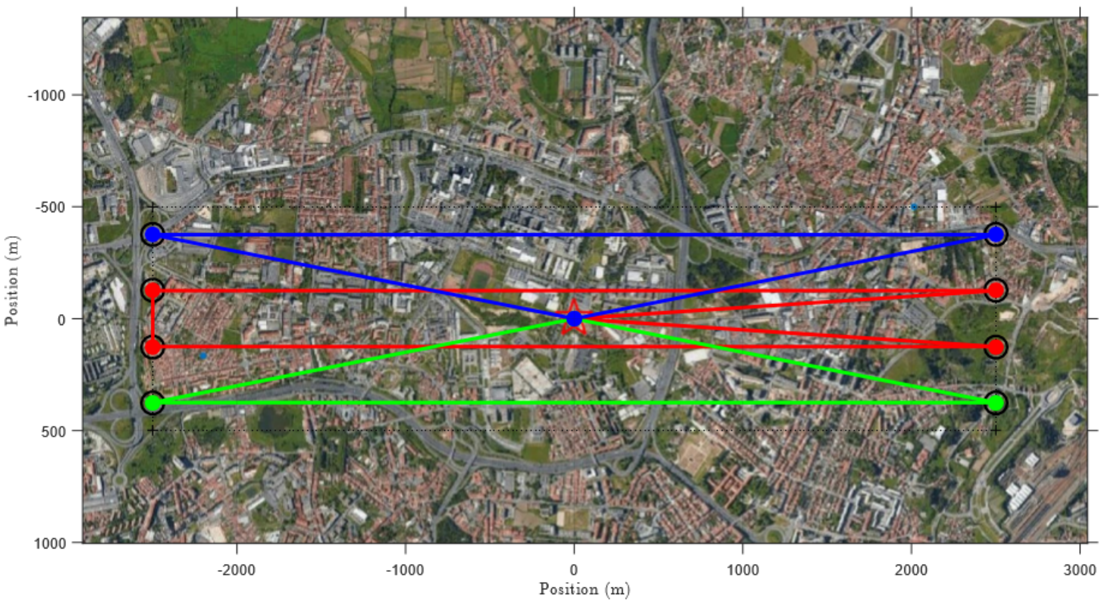
generally lower flight speed due to weak aerodynamics design.

For comparing each one is the best type we performed a general test, where we would apply the same number of operators and drones to cover the same area, changing however the constants that differs on each drone specification accordingly to Tables 6.1, 6.2 and 6.3 [15].

In Fig 6.12 we see the comparison between the two types. In the first simulation, presented in Fig. 6.12a, we obtain the partial times corresponding to red drone's route, green drone's route and blue drone's route, with times of 10 minutes and 47 seconds, 7 minutes and 5 seconds and 7 minutes and 3 seconds, respectively. Resulting in a total mission time of 13 minutes and 3 seconds for the fixed-wing drones. In the same way, Fig. 6.12b shows the resulting routes for the red drone's route, green drone's route and blue drone's route, where partial times are 15 minutes and 16 seconds, 10 minutes and 12 seconds and 10 minutes and 13 seconds respectively. Thus the total time of the mission when using a team of quadrocopters is 24 minutes and 9 seconds. Which is a big difference when comparing to the fixed-wing.



(a) Coverage Method using a team of 3 x8 skywalker accordingly to Table 6.3 values.



(b) Coverage Method using a team of 3 Mariners accordingly to Table 6.3 values.

Figure 6.12: Comparison of the system overall performance for two distinct type of drones.

Despite the fixed-wing appear to be largely recommendable for this area coverage method, further considerations for more realistic approach need to be made that will decrease the gap between the two drone types. For smaller and sharper turn areas, like non-convex polygons the quadrocopter performs well but for wide areas with large turns, fixed-wing will be a wise choice.

Moreover, we conclude on the several executed simulations that the method from [17] is applied to rotate the polygon into the right angles and then the rows are plotted always parallel to polygon largest side, and perpendicular to the sweeping direction movement.

In conclusion we showed on these simulations how the different constraints imposed over the area coverage method impact the final solution. Also, we saw that in this method the constant values assumed may, in specific situation, make the solution infeasible or not global optimal. Despite precise in some practical issues, this algorithm is still very incomplete regarding issues like the battery wasted from departing and returning to depot point, fault monitoring, area coverage geometric forms among others.

6.2 Communicative area coverage

The simulations made for the distributed communicative area coverage algorithm represent the solution to the problem presented in chapter 4 using the methodology presented in 5, based on the works of [16] and [19].

The exposure of this method will be showed in several simulations. On the first ones, we will examine the behavior of the system when the importance weight associated to each point in the region of interest its not uniform, this is, there are regions with higher priority than others. Then, we will survey the reaction of the system when different areas are imposed, and try to highlight based on equations aforementioned in chapter 5 the most important issues. Furthermore, we will show how the coverage quality function evolutes and how the area covered percentage reacts to the change of sensed region footprint. For last, we will show how the system reacts to malfunctions of one or many drones. Lastly, we will increase the simulation time to see the effects on the overall performance of image coverage quality and total area covered.

Note that, apart from non-communicative area coverage method presented, this method works on outdoor or in-door. Once the drones system coordinates is made through their communications fields. We also remind that, in this method, we assume that the nodes are already inside the regions on random positioning or not.

Simulation results of the proposed method [16] using the uniform coverage quality function q is presented along this section. All drones are homogeneous, with a half sensing cone angle of $a = 20^{\circ}$ and $z_i \in [0, 2.3], \forall i \in I_n$. The boundaries are shown in solid black and the boundaries of the sensing disk in dashed red lines. The time of simulation was of 15 seconds which is enough to study the behavior of the area coverage method.

6.2.1 Initial points

One solution if we want to minimize the cells overlapping on the initial steps is to execute some traditional static multi-agent system area coverage algorithm, that uses voronoi tessellations, for instance. One proposed method if we want the drones to start already on the expected theoretically positions is to execute a cluster algorithm like the one shown in Fig. 6.13. In this figure, a algorithm based on Lloyds clustering is used to, given a certain area, distribute the agents on the centroid of the generated voronoi cells.

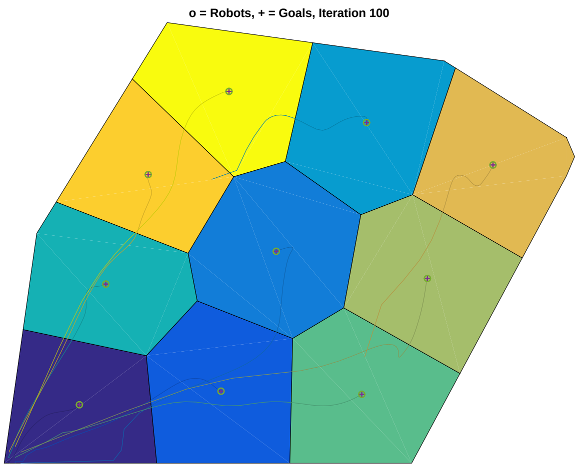


Figure 6.13: MATLAB Simulation of a Lloyds's clustering based on voronoi limits algorithm for deployment of a region spatial optimized initial distribution of a given number of agents among a given area.

This approach allow us to have more optimized initial positioning for the communicative area coverage algorithm. Also the Fig. 6.13 presents a static communicative coverage method. Drawback of this method is that once the drones are on the centroid of voronoi cells they wont move any longer. Not compensating in case of drones malfunctions. Opposite to the method we are going to simulate further.

6.2.2 Space density variation Impact

In our first tests, we consider two different simulations where the importance associated with each point within the area A is constant and when is not constant, this is, the importance of each regions to be covered is not the same and follows a Gaussian distribution. For each of one of this simulations we will see how the number of drones impact the final result. Given the area defined in Fig 6.13.

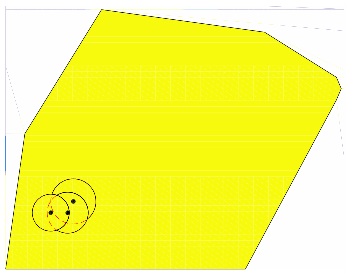
In first simulation, we assume that we have 3 drones to cover the area, which is not enough to cover all the area. Thus, a simulation was ran where, drones distribute on the area with $\phi(f) = 1$, seen trough the simulation in Fig. 6.14. Where Fig. 6.14a and Fig. 6.14b, it is the drones initial location and the drones final location, respectively. The yellow color means that the importance is the same in every region. This way, due to the fact that that three drones is not enough to cover all the area as shown in Fig. 6.14b, they will go for the optimal altitude as seen in Fig. 6.14g, and in Fig. 6.14c to Fig. 6.14c, being this the best local solution and configuration as shown in Fig. 6.14e, however we see that the objective function could be higher by watching Fig. 6.14f. Despite, the drones coverage could be higher, for higher altitudes, the coverage quality function would be lower and the system always look for the best configuration between area coverage and image coverage quality. Result is an area coverage of approximately 45% as seen in Fig. 6.14h.

In second simulation, we assume that we have three drones in the same situation of last simulation, however this time the region has $\phi(f) \neq 1$. Thus, a simulation is ran where, drones distribute on the area with $\phi(f) = GaussianSurface$, seen through the simulation in Fig. 6.15, where Fig. 6.15a and Fig. 6.15b, it is the drones initial location and the drones final location, respectively. The mesh of colors means that the importance is not the same in every region. This way, due to the fact that three drones is not enough to cover all the area as shown in Fig. 6.15b, they will go for the stable altitude as seen in Fig. 6.15f, and in Fig. 6.15c to Fig. 6.15c, similarly to last simulation, being this the best local solution and configuration as shown in Fig. 6.15e, however we see that the objective function could be higher by watching Fig. 6.15e. Despite, the drones coverage could be higher, for higher altitudes, the coverage quality function would be lower and the system always look for the best configuration between area coverage and image coverage quality. Result is an area coverage of approximately 45% as seen in Fig. 6.15g but with the region with more interest almost fully covered, boosting the objective value function when compared to the last simulation example.

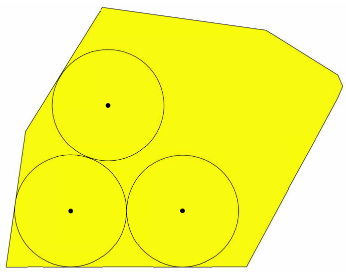
We see that in this case, the drones tend to occupy first the most relevant region in the area and then maximize the coverage. The area was not covered as it could because it was benefic for the network not to go for optimal altitude. Instead of that its stay a bit under that altitude and obtain a better image quality of the region that truly matters. The jitter observed between the cells of some drones happens due to the arbitrary assignment of boundary of one of those drones. Once the altitude of one drone changes slightly, the boundary between their cells will change instantaneously from a line segment to a circular arc. However, quality function will not present any discontinuity face to this problem [16]. In those last cases, the system achieved its maximum local optimality once the sensed cells on the region are at its highest value from coverage function. The fact that the covered area increases monotonously is no surprise since the control law was designed so that the joint coverage-quality objective criterion increases monotonously, thus there is guarantee that the covered area will also increase monotonously, with or without the use of a space density function $\phi(f)$.

6.2.3 Influence of large networks

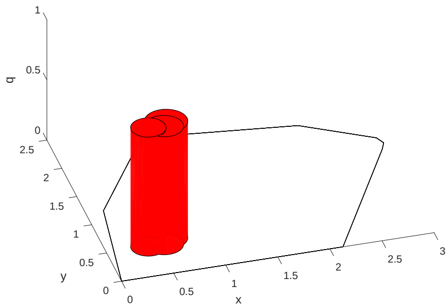
The second tests, based on last two simulations, aims to explore the network evolution of the area coverage method with homogeneous importance region of interest, for a team of 9 drones. From



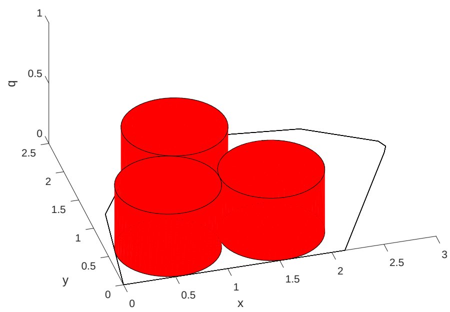
(a) Drones initial location.



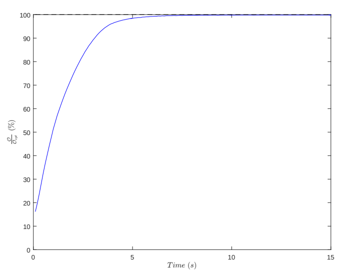
(b) Drones initial location



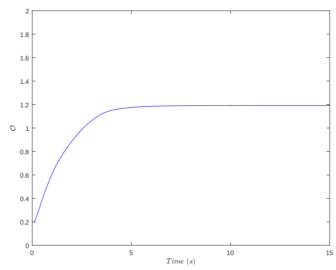
(c) Initial Quality Value.



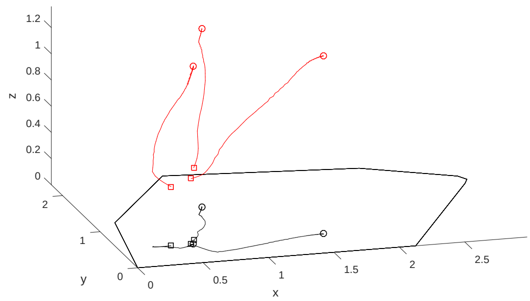
(d) Final Quality Value.



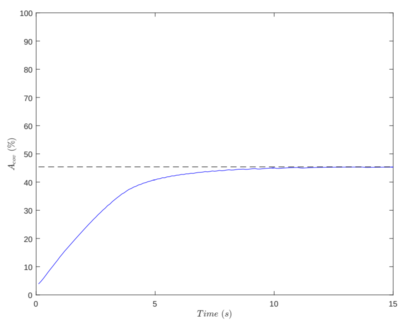
(e) Objective Function relatively to the possible best solution.



(f) Objective Function.



(g) Drones Trajectories, squares are initial positions and circles the final positions.

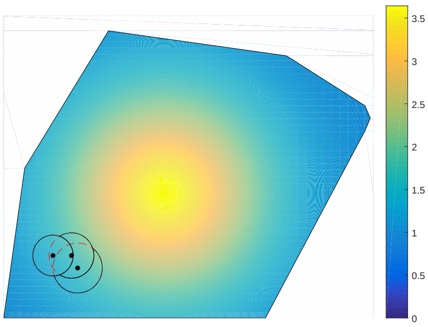


(h) Area covered in this configuration.

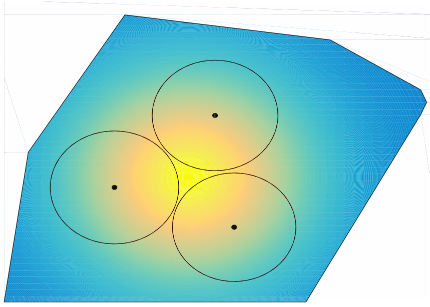
Figure 6.14: Simulation of a 3 drone system with homogeneous region importance.

Fig. 6.5 we see the system evolution for 9 drones randomly disposed initially, with $\phi(f) = 1$ and their repercussions on system performance.

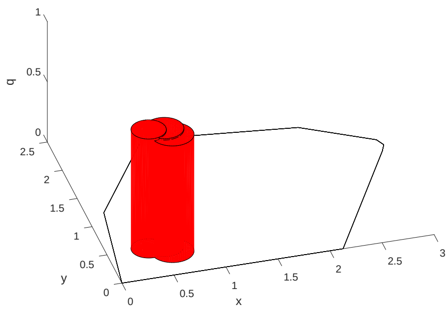
From this simulation we can see that the holes and gaps between the cells and the region boundaries have an impact on area coverage percentage, which is approximately 88% when it



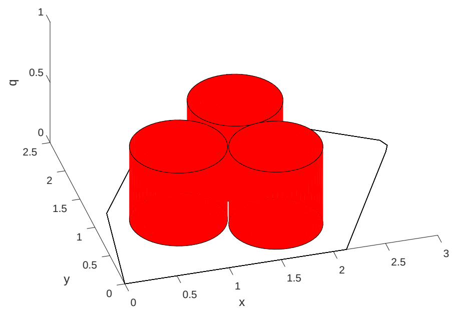
(a) Drones initial location.



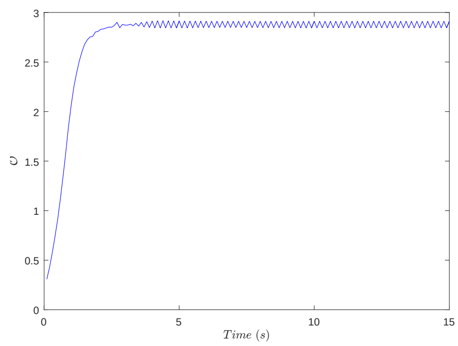
(b) Drones initial location



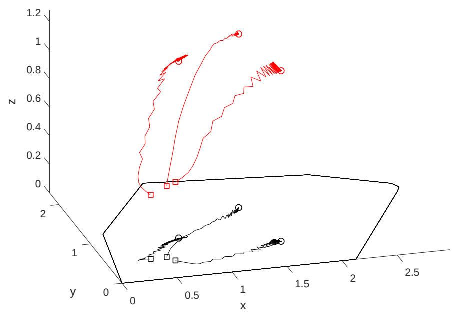
(c) Initial Quality Value.



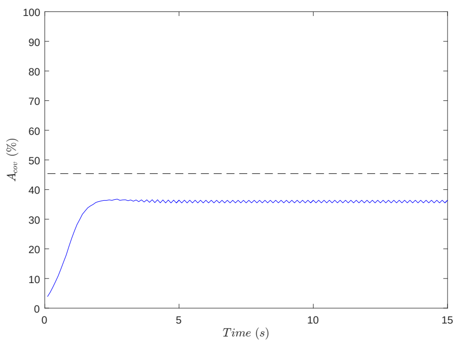
(d) Final Quality Value.



(e) Objective Function.



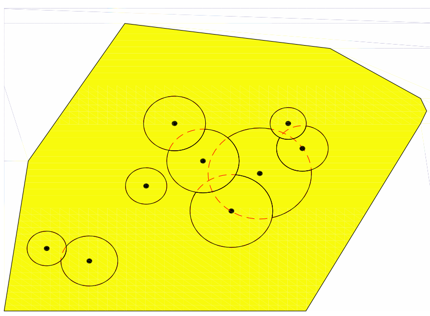
(f) Drones Trajectories, squares are initial positions and circles the final positions.



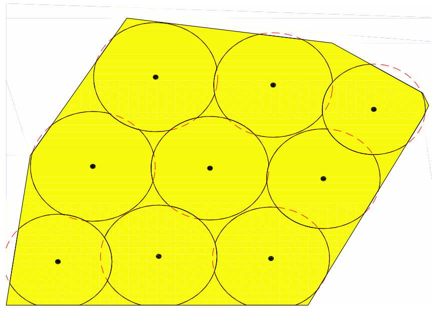
(g) Area covered in this configuration.

Figure 6.15: Simulation of a 3 drone system with equal region with heterogeneous importance.

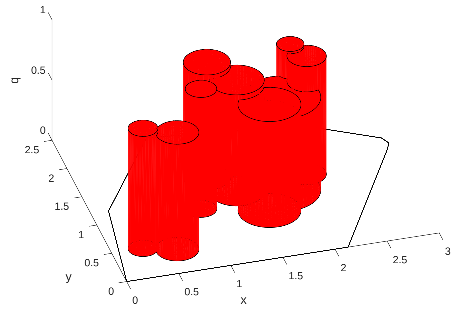
should be 100%. This happens because if drones sensed region was fully covered image overlapping would happen and the system would give privilege for the drone with better image quality,



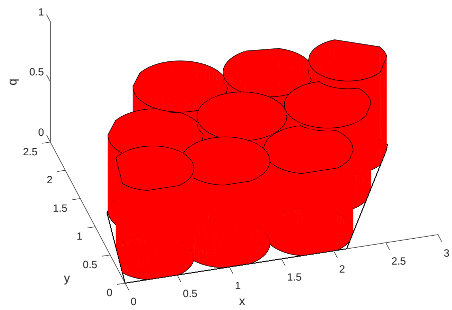
(a) Drones initial location.



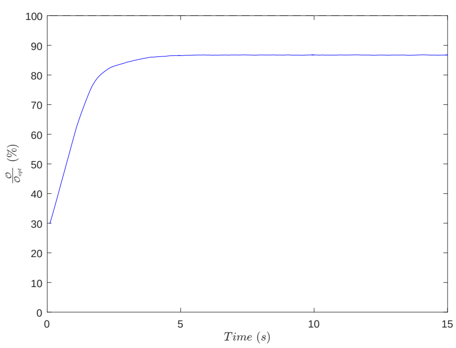
(b) Drones initial location



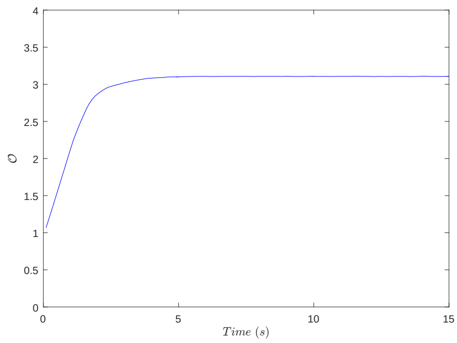
(c) Initial Quality Value.



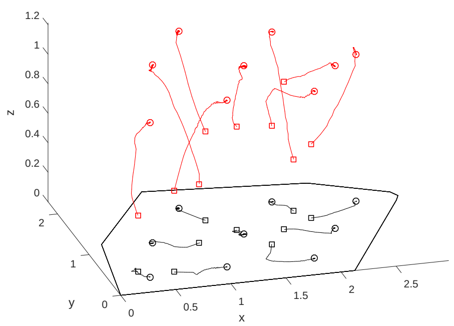
(d) Final Quality Value.



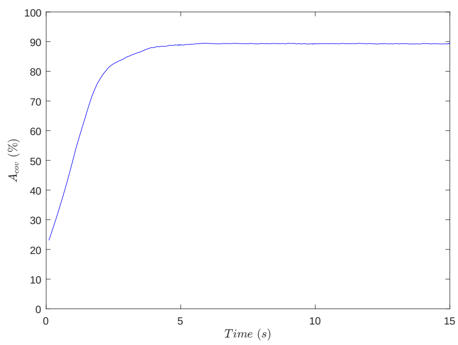
(e) Objective Function relatively to the possible best solution.



(f) Objective Function.



(g) Drones Trajectories, squares are initial positions and circles the final positions.



(h) Area covered in this configuration.

Figure 6.16: Simulation of a 9 drone system with homogeneous region importance.

neglecting some drones and getting even worst coverage percentage and quality coverage function value. From Fig. 6.16g it can be seen from the trajectories that the altitude of some nodes was not constantly increasing. This is expected behavior since drones at lower altitude will increase the stable altitude of drones at higher altitude they share sensed regions with. Once they no longer share sensed regions, or share a smaller portion, the stable altitude of the upper node will decrease, leading to a decrease in their altitude.

In same situation as in simulation of Fig. 6.16 it is expected that, for a Gaussian surface, the network behavior would follow the same issues has the simulation of Fig. 6.15 but this time applied to a team of 9 drones just like Fig. 6.16.

From these simulations we conclude that for uniform and changeable Gaussian surfaces, the system global optimal value is not proportional to the number of drones, once in large number they turn off the ones with poor coverage quality, which is not bad if we think they can substitute on drone in case of malfunctions. Beside, the area percentage covered in this algorithm's methodology [19] [16] is intrinsic to the region shape and sensed region pattern.

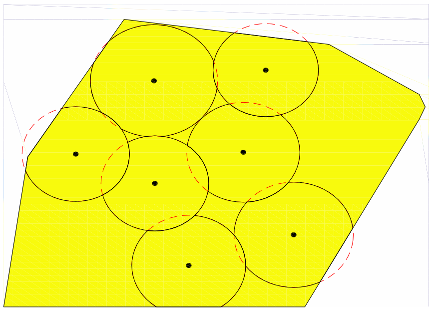
6.2.4 Fault Detection

Based on the same 9 drones team used in the simulation of Fig. 6.16 we withdraw the drones final locations present in Fig. 6.16b and remove two random drones from the network, to simulate a malfunction situation and we examine on further simulation what is the network next step to enhance the quality coverage quality and area coverage's percentage. In Fig 6.17 we present the simulation results from deleting 2 drones from the 9 drones last presented network. This simulation showcases the robustness of the proposed control law to the removal of drones. For this purpose, two nodes are removed from the final configuration of Fig 6.16 simulation, resulting in the initial configuration seen in Fig. 6.17a. The network converges to the final configuration seen in Fig. 6.17b. In Fig. 6.17g we see the drones location evolution. We also see that in Fig.6.17d the objective function reached a new local optimal configuration, without reaching the global possible maximum. Since the region is not big enough to contain seven sensed disks, the drones converged to altitudes lower than the optimal, as expected.

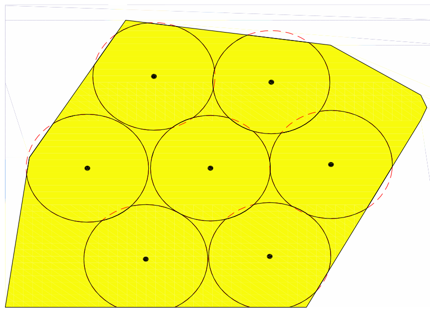
In this simulation we show one of the biggest features of this system which is not only the real time area coverage with continuously improving coverage image quality but also the fail monitoring and compensation which drives the system for always a local optimum value and making this area coverage method a robust method for surveillance applications, even the tough ones like the military scenarios.

6.2.5 Impact of area and different sensing patterns

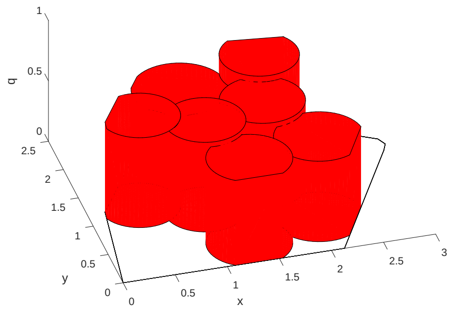
As mentioned before, the coverage quality function is intrinsic related to the area to be covered and in the sensing pattern. The holes and gaps in the coverage exists due to the fact that we avoid for cameras of the drones to overlap their images because would result in loss of coverage quality. Also, sometimes the geometry between the area and the different drones are not compatible and



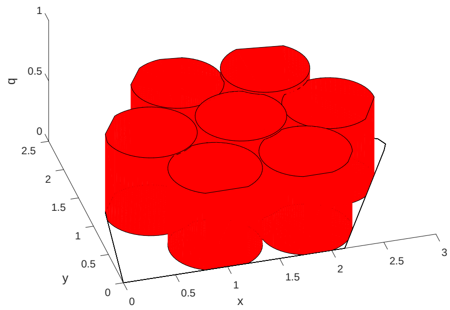
(a) Drones initial location.



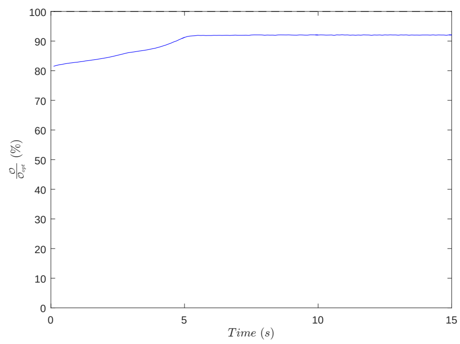
(b) Drones initial location



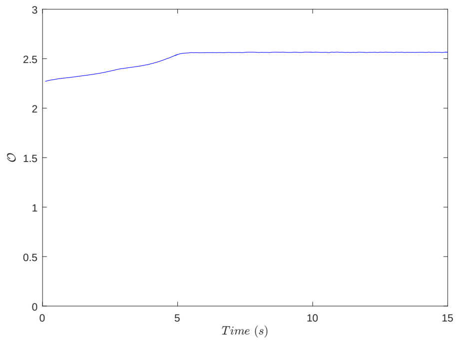
(c) Initial Quality Value.



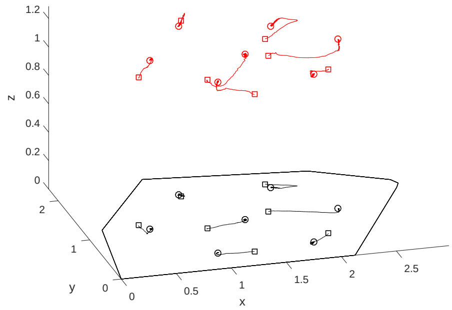
(d) Final Quality Value.



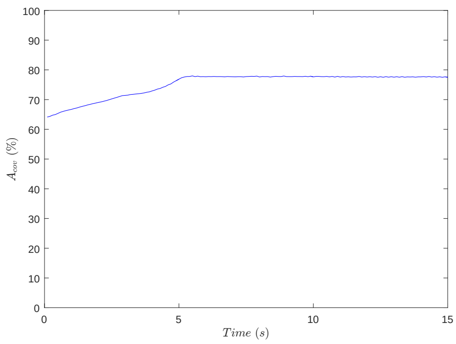
(e) Objective Function relatively to the possible best solution.



(f) Objective Function.



(g) Drones Trajectories, squares are initial positions and circles the final positions.



(h) Area covered in this configuration.

Figure 6.17: Simulation of a 7 drone system with homogeneous region importance.

therefore results in holes in coverage. Through Simulations made in this section we try to illustrate the behavior of a different areas shape and different sensing patterns in overall performance. Note that, for the case where the sensing pattern are not disks it is better if we consider drones orientation variables in order to enhance the global results of the coverage method. Considering for this case the work of [19], [121].

To inspect the impact of these changes on the method proposed in chapter 5 we consider the new area to be square when needed, and the sensing pattern to be square when needed. To approximate this methodology from real scenarios we assumed rectangular cameras footprints over the planar region and regions to be covered with rectangular shape as well. Also considered a team of 8 drones.

In first simulation we changed sensing pattern keeping the area shape. The results can be seen in Fig. 6.18 plots. In second simulation we change the area shape but keep the sensing pattern as a disk. The results from this simulation can be seen on Fig. 6.19. For last we simulate with different area shape and different sensing pattern from last both simulations. The result is shown in Fig. 6.20 plots.

In this first simulation, we see that in the beginning the drones moved very quickly to a equal space distribution (From Fig. 6.18a to Fig. 6.18b). However, we see from their trajectories Fig. 6.18f that an equilibrium was not found to the angular nature of the sensing pattern and also the area shape. Although, the algorithm was driven to an optimal local solution Fig. 6.18d. Despite all the jitter due to several cells collisions, the area coverage was surprisingly pleasant of 75% (Fig. 6.18g). If carefully controlled could have an higher percentage.

In second simulation, where the area shape was considered rectangular presented in Fig. 6.19, the result are better if the same amount of drones was applied to cover more angular shaped area once the sensing patterns cells avoid to stay out the region of interest, because does not lie any optimization of the method's algorithm in there. Therefore, for sharp areas, the drone will lower his altitude to cover one small area or, would simply not cover, becoming that region a gap or a hole in the covered area. We can also, conclude by the final disposition of drone in Fig. 6.19b that with one more drone the area would be very closed to 100% coverage. Its also visible that the cells are at his maximum values for the quality cover function, once the relative objective function its on his local maximum configuration, as seen in Fig. 6.19d and Fig. 6.19e. It also perceptible that angular sensing patterns cause the system to have much more jitter than when considering rounded patterns. This can be proven by comparing objective function's monotony in Fig. 6.18e and Fig. 6.19f.

As seen in last simulation, where area's shape and sensing pattern are considered rectangular, as shown in Fig. 6.20, and one more extra drone to completely cover all the area is also considered. That if the nodes are placed carefully on favorable initial locations Fig. 6.20a, the area will be almost 100% covered in Fig. 6.20h. Leading to one of the best possible area surveillance mechanisms. We can also see the values of the objective function in Fig. 6.20e and 6.20f, and the final cylinder volume, which means the quality coverage integrals values, in Fig. 6.20d.

We show asset on last simulations that with this method, despite avoiding cells collisions and image overlapping. We also increase the area covered and quality coverage function values. So one can conclude that different sensing patterns and area's shape have big implications on the overall network performance and configuration.

However, beside all advantages that this method offers to the user to surveil the area, also has its drawbacks, one of them is the aggressive direction changes, which would need to implement the real drones kynodynamic models in order to deny such aggressive movements. This algorithm its also energy expensive, once the drones are continually moving. Beside, the algorithm should incorporate a base or multiples bases where drones could recharge, take off and land. Drones crashes it is also an issue in this method once the altitudes range is predetermined. This is a problem because this method is specially useful for area coverage in hostile scenarios where we usually do not have any infrastructure, relying on drone *peer-to-peer* network communication. One solution is to implement on the algorithm a sense and avoid method.

6.3 Comparison

Comparing the two algorithms proposed in this chapter 6 is quite difficult once we are dealing with two total different methods. However, from the drawbacks of each one of the methods and their advantages, we can make some predictions.

Non-Communicative method is perfect when the area to be covered is not time-critical and we only want to check once in a while whats happening. This is a perfect solution for agricultural purposes like crops monitoring. Also, this method is generally for big areas, and where the level of hostilenesss is low or none. We saw that this method is normally boosted if FWUAV drones are used. The drawbacks of this method is that in case of drones malfunctions we cannot simply cover the area like we should, having big coverage loss.

On the other hand, communicative coverage, like the one in this method, is optimal for hostile scenarios. The drones can surveil areas with more interest that others and prioritize certain regions inside the area of interest by using Gaussian Surface Distribution. Other advantage of this method is the fact that its a dynamic model which means that they can cooperative perform the tasks and assume command of missions when other drones are with malfunction or away. Also, they can have different altitudes. Allowing better image quality or more area coverage depending on the mission requirements. The aggressive maneuvers required by this method makes it hard for being used on FWUAV so the rotary-wing drones are the more suited ones on this method.

6.4 Conclusion

In this chapter we simulated the methodologies aforementioned in Chapter 5.

We started by specifying the specifications common ground for the non-communicative simulations. Based on [15] knowledge about drones, was established a comparative term between the two different types of drones. Their specifications were plotted and the simulations characteristics defined by them. Then, we explored the algorithm robustness by eliminating some constraints of the method for non-communicative area coverage. Proving that the methodology presented in Chapter 5 Section 4.1 was well design. We started for testing the Objective function, followed by constraints that did not allowed the problem to be a TSP problem but a coverage problem. Moreover, we tested constraints were problem complexity was minimized despite the increase of the minimum time to perform the routes.

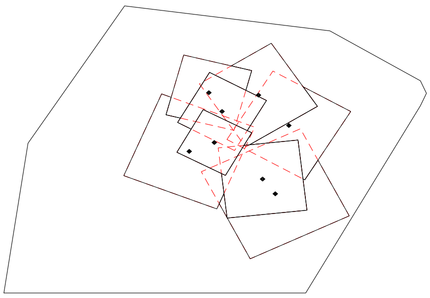
After, other situations were took into account like the setup time when we only have one operator. Showing that this method not only gives us the optimal solution but also knows when its not worth to use all the drones available. Then we examined the methodology drawbacks. Starting with big optimization loss due to the vehicle endurance. We also, explored what would happen if the system had malfunctions like one drone crash. Other drawback exposed was the coverage redundancy coming from non-convex areas coverage, because the method transforms it into the convex hull of that non-convex polygon, wasting time and energy on regions with no interest at all.

For last, we explored the space partitioning through the variation of the method constants, like altitude, flight speed or even image overlapping. In final approach to the non-communicative algorithm we compared which one of the drone types aforementioned best suits the method.

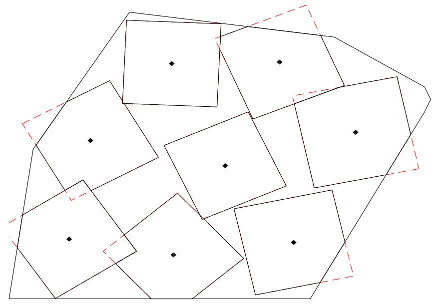
For communicative area coverage we began with a brief introduction of what it was defined in Chapter 5 and 4. We then outlined the simulations procedure and general specifications.

We introduced one solution to attribute in the method simulation an optimal initial drones location using a Lloyd's Voronoi based cluster algorithm. After that, we explored for this area coverage method, the impact of the variation of the importance of certain regions within the area and made some simulations and tests to see the results from it. Then, we studied the behavior of this method for a given area with large number of drones and saw where the solution would converge in terms of planar positioning and altitude due to the control laws nature. Then we exploited one of the biggest features of this method which is the capability for, in case of missing drones, to compensate their work, redistributing among the area. For last, we simulated the area coverage assuming that the area's shape was different and the sensing pattern as well. Resulting in some cases on better area coverage and optimal system configuration.

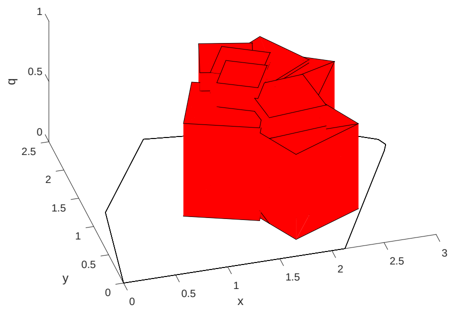
To conclude we briefly compared the two methods presented in the light of their general drawbacks and advantages and which type of drones team was more appropriated and for which scenario.



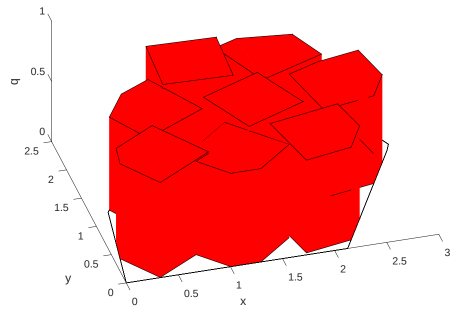
(a) Drones initial location.



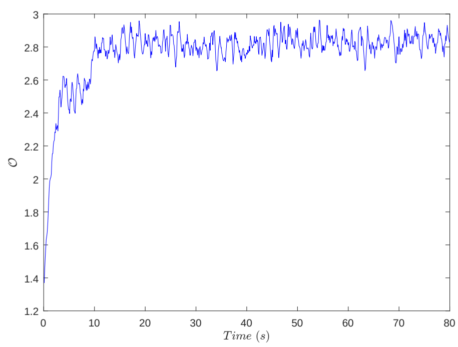
(b) Drones initial location



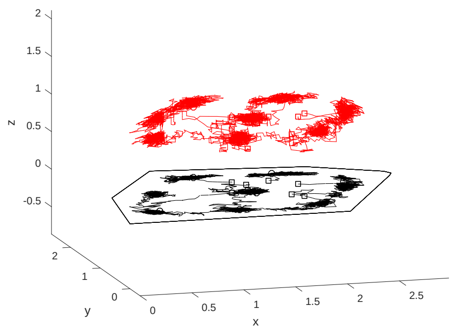
(c) Initial Quality Value.



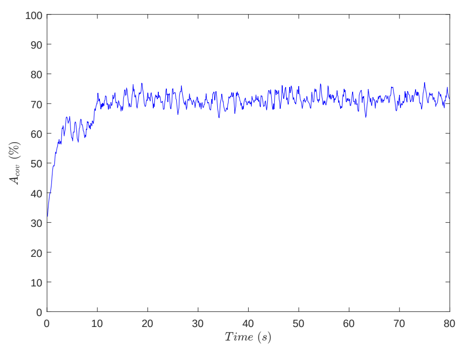
(d) Final Quality Value.



(e) Objective Function relatively to the possible best solution.

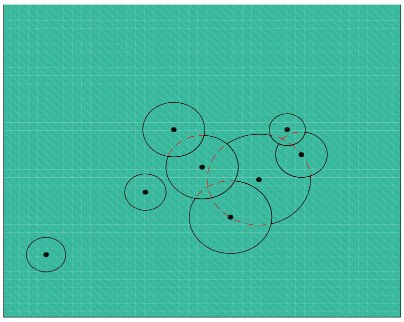


(f) Drones Trajectories, squares are initial positions and circles the final positions.

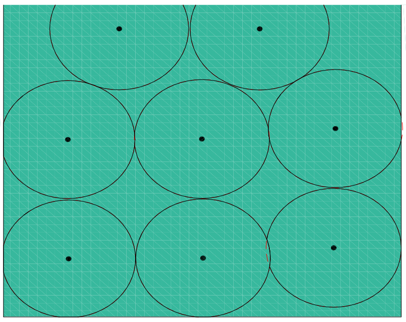


(g) Area covered in this configuration.

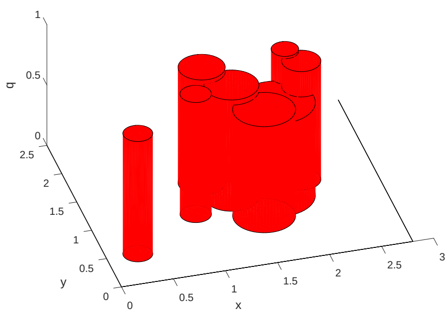
Figure 6.18: Simulation of a 8 drone system with homogeneous region importance and rectangular sensing pattern.



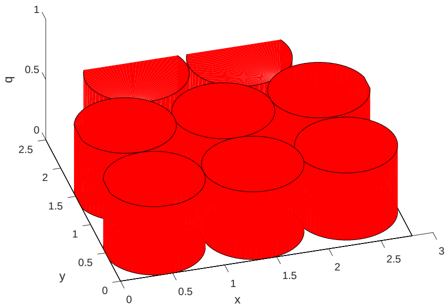
(a) Drones initial location.



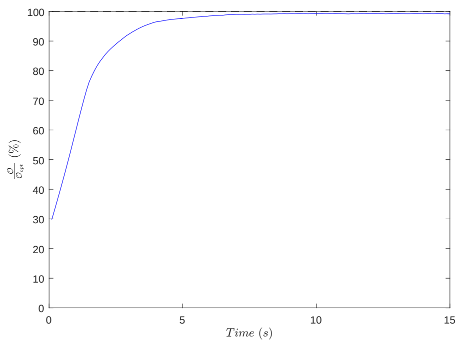
(b) Drones initial location



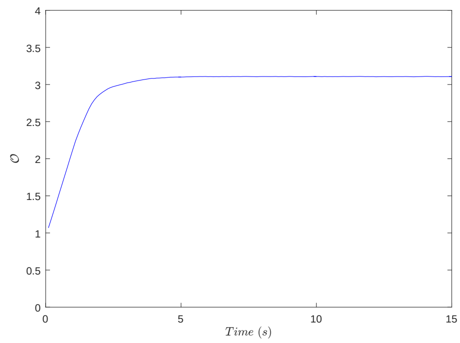
(c) Initial Quality Value.



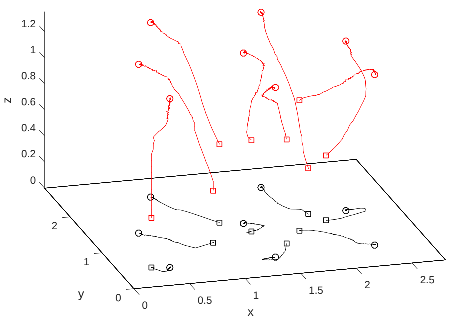
(d) Final Quality Value.



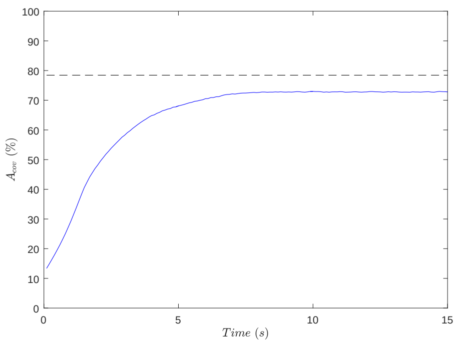
(e) Objective Function relatively to the possible best solution.



(f) Objective Function.

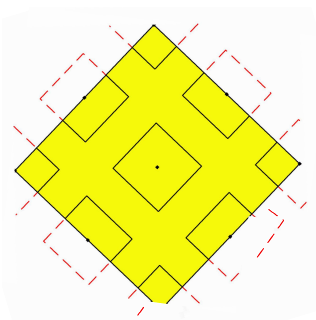


(g) Drones Trajectories, squares are initial positions and circles the final positions.

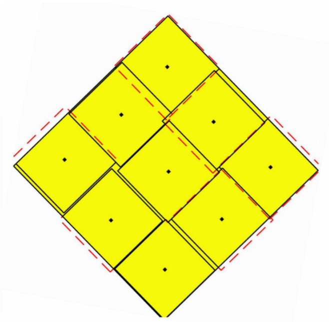


(h) Area covered in this configuration.

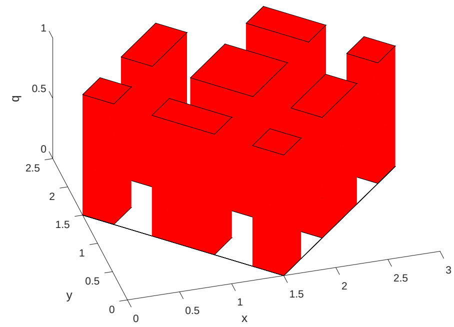
Figure 6.19: Simulation of a 8 drone system with homogeneous region importance and rectangular area shape.



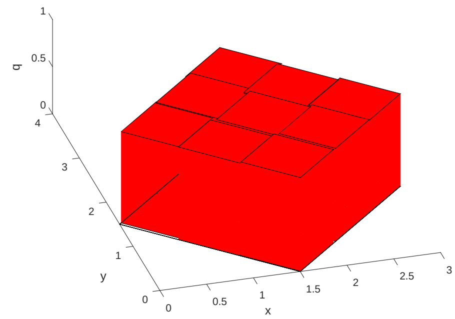
(a) Drones initial location.



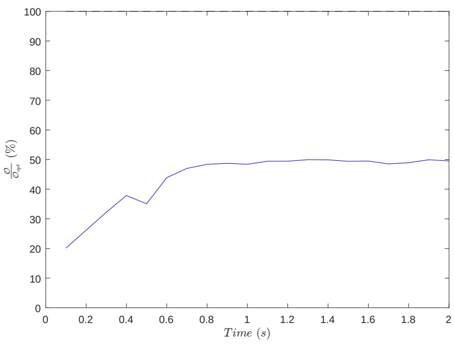
(b) Drones initial location



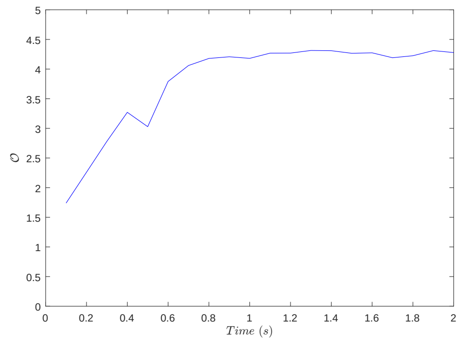
(c) Initial Quality Value.



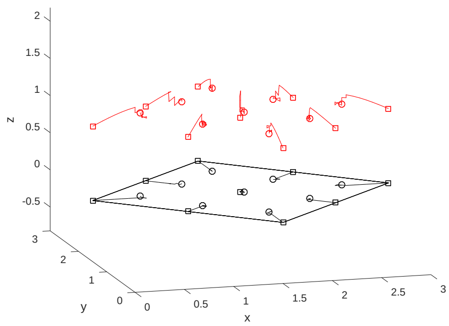
(d) Final Quality Value.



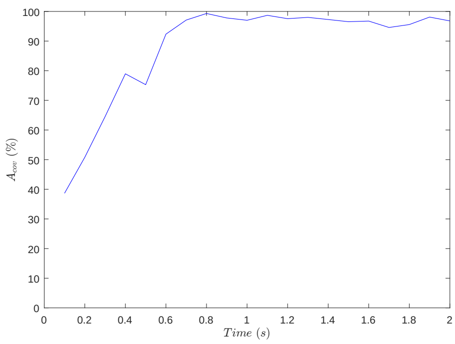
(e) Objective Function relatively to the possible best solution.



(f) Objective Function.



(g) Drones Trajectories, squares are initial positions and circles the final positions.



(h) Area covered in this configuration.

Figure 6.20: Simulation of a 9 drone system with homogeneous region importance, rectangular area's shape and rectangular sensing pattern.

Chapter 7

Conclusions and Future Work

This thesis has presented different solutions for area coverage problem. The algorithms are divided into non-communicative and communicative area coverage. In non-communicative, the drones perform the assigned tasks disjointly by sweeping over the region of interest. They follow optimized routes assigned by a base station and departing from there. In communicative coverage, the agents start at an initial random location on the border of the region or already inside, and then, spread dynamically along the region dividing as much as possible the responsibility of the covered area. In this coverage, the drones cover regions within their sensing range and are able to communicate with each other as long as they are inside of each other communicative area. This *peer-to-peer* communication turns this method into a MANET, creating a distributed drones system. This last coverage methods allows the area to be monitored longer time and in near real-time. However it requires much more agents when compared to the non-communicative coverage.

In this thesis, we started by introducing the necessity of this works and its contributions to the modern state of surveillance technologies. Then we shared a detailed background knowledge about UAS, mentioning their history, hardware and software differences, their classification accordingly to different points of view and many system components. Additionally, we exposed some of the flight control mechanisms, the general flight dynamics and how drones autonomous levels are defined. Moreover we cite some applications on the many different fields where drones take major roles nowadays. For last in that chapter, we introduced the concept of multi-agent systems and how they could operate with each other.

Posteriorly, we explained based on many authors and literatures, the pioneer approaches in which our work assets was based on. Defining what surveillance was, in the context of this thesis, and what area partitioning and coverage algorithms exist. Scoping, for last, the two desired maneuvers for an area coverage problem, the non-communicative and the communicative coverage. As last, we briefly explained the works done by [63], [88], [18] and [16].

After in this thesis, we explained the problem itself, what was the purpose and what assumptions were made to solve it. In this case, we assumed a globally area coverage problem in which, different approach had different assumptions and requirements.

Finally, we explained in detail the algorithms adopted to solve each one of the problems imposed, the non-communicative offline path planning area coverage methodology and the communicative cooperative visual area coverage methodology, respectively.

For last, we implemented both scenarios algorithms on MATLAB and tried to validate the methodology adopted through a bunch of tests such as robustness tests on the methods bottlenecks or optimized results incoming from methodology adopted, comparing to traditional solutions.

7.1 Objectives Satisfaction

In terms of objectives satisfaction, we were able to complete most of them, leaving however behind some implementation of these methods into a more realistic simulation platform beside from MATLAB, testing this implementation for Software-in-the-loop toolchains and even try to validate some methods with real field experiments or Hardware-in-the-loop executions.

The proposed initial objectives were to adopt and optimize area coverage surveillance techniques and implement them on LSTS [15] toolchain, And, through tests and controllers adjusts, optimize the already implemented area coverage algorithm.

Related to drones general knowledge, we were able to gather a lot of information from other authors creating a nice research summary and review over drones surveillance methods and applications. From literature review, we were able to scope the bigger authors for each pioneer work and further developments. Also, satisfied the need of finding and design methods to surveil areas using different technologies, in this case communicative and non-communicative. For last, we were able to implement on MATLAB the different algorithms as desired for first simulations and draw some conclusions over the studied methods. Presenting, this way, two coverage methods for two very distinct and appealing procedures.

To be done, was half of the interface that would connect our MATLAB algorithms directly into LSTS toolchain, allowing us to make Software-in-the-Loop Simulations. Using Neptus as the mission planer, IMC as the communication interface and DUNE as the vehicle on-board controllers. Validating in real life the methods adopted by adding real life constraints like wind speed and direction or even battery usage.

7.2 Future work

In terms of future work this thesis can be divided into three parts.

In the first part, by improving and optimize even further the non-communicative coverage algorithm. Some suggested improvements are:

- Incorporate the drones kynodynamic models into the simulation so we can precisely know what are the accelerations, decelerations and turn angles need to be done to minimize paths time and energy consumption;

- Create a fault monitoring system where the operator could deploy another extra drone to perform the non covered area, this can be achieved by using some kind of communication in this method;
- Improve the space partitioning so we can divide non-convex polygons into several convex polygons decreasing mission total time and coverage redundancy [17];
- Make the algorithm modular for any coordinate system, facilitating the export of algorithm information into other platforms;
- Create a situation where the UAV could be deployed from many different depots;
- Introduce other environment variables like the wind speed, or the compensation on sweeping direction in opposition to wind direction impacts.
- Introduce Vertical movement optimization;
- Consider the decreasing battery variable in face to environment conditions and the consumption when departing to base station or returning.
- For last, create an incorporated collision avoidance system so we make sure that the UAV do not crash each other or in anything on the terrain.

In the second part, by improving and explore even further the communicative coverage algorithm. Suggested improvements are:

- Create a complementary algorithm were drones fly from a base or several bases to the intended region;
- Create a coordinate system so we can plot the algorithms results in a more realistic environment;
- Add to the algorithm the real vehicles kynodynamic models throw an interface, so we can take real time values;
- Explore the communication issues like communication delay, ad hoc algorithm, among other communications general problems;
- And finally, implement in an advanced level some surveillance features like surround a given target, for example if we define our Gaussian surface variable in time, we can realize intruders on the area, with image recognition, or increase patrol over areas with more probability of being problematic.

And for last part and the most important one, to:

- Improve and add features to MATLAB interface with DUNE, IMC and Neptus so we can directly export the algorithms data to LSTS toolchain and run SITL and HITL tests;
- Validate this algorithms in may different Autopilots, drones and networks so this algorithm can be improved to a satisfying level.
- Explore and add another features apart from area coverage used for surveillance technology, remembering that surveillance has many applications for military scenarios but also many commercial applications. Some suggestion are, camera's image recognition patterns, self-learning teams of drones, search and destroy or search and rescue algorithms, Mapping among other technologies;
- Create a program or a framework, where any kind of drone could use or simulate this algorithms, for future use by the LSTS or even for commercial uses.

References

[1] Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems, 3 2012. URL: http://doi.wiley.com/10.1002/rob.20414, doi:10.1002/rob.20414.

[2] Chan Woei-Leong and Hsiao Fei-Bin. Implementation of the Rauch-Tung-Striebel smoother for sensor compatibility correction of a fixed-wing unmanned air vehicle. *Sensors*, 11(4):3738–3764, 3 2011. URL: http://www.mdpi.com/1424-8220/11/4/3738/, doi:10.3390/s110403738.

[3] De Havilland Tiger Moth / Queen Bee | BAE Systems | International. URL: http://www.baesystems.com/en/heritage/de-havilland-tiger-moth---queen-bee.

[4] Radioplane OQ-14/TDD. URL: http://www.designation-systems.net/dusrm/app1/oq-14.html.

[5] M Dempsey. Us army unmanned aircraft systems roadmap 2010-2035. *Federation Of American Scientists*, pages 1–140, 2010. doi:ATZQ-CDI-C.

[6] M Hassanalian and A Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Prog in Aerosp Sci*, pages 1–33, 2017. URL: http://ac.els-cdn.com/S0376042116301348/1-s2.0-S0376042116301348-main.pdf?_tid=25adc3a4-4ce4-11e7-9ad7-00000aacb35e&acdnat=1496993022_4507e6d2009a6a10a279968814710357http://linkinghub.elsevier.com/retrieve/pii/S0376042116301348, doi:10.1016/j.paerosci.2017.04.003.

[7] Adam C. Watts, Vincent G. Ambrosia, and Everett A. Hinkley. Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing*, 4(6):1671–1692, 2012. doi:10.3390/rs4061671.

[8] CCRP - Command and Control Research Program. URL: http://www.dodccrp.org/html4/journal_v1n2_01.htmlhttp://www.dodccrp.org/html4/journal_main.html.

[9] Richard S Stansbury, Manan A Vyas, Timothy A Wilson, R S Stansbury, T A Wilson, and M A Vyas. A Survey of UAS Technologies for Command, Control, and Communication (C3). *J Intell Robot Syst*, 54(54):61–7861, 2009. URL: http://download.springer.com/static/pdf/448/art%253A10.1007%252Fs10846-008-9261-2.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Farticle%2F10.1007%2Fs10846-008-9261-2&token2=exp=1497373853~acl=%2Fstatic%2Fpdf%2F448%2Fart%25253A10.1007%25252Fs10846-008-926, doi:10.1007/s10846-008-9261-2.

[10] Kimon P. Valavanis and George J. Vachtsevanos. *Handbook of unmanned aerial vehicles*. 2015. doi:10.1007/978-90-481-9707-1.

[11] a Noth, S Bouabdallah, and R Siegwart. Dynamic Modeling of Fixed-Wing UAVs. *Master Course in Mechanical Engineering*, pages 0–11, 2006.

[12] Markus Hehn and Raffaello D Andrea. Quadrocopter Trajectory Generation and Control. *IFAC World Congress*, 18(1):1485–1491, 2011. URL: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac11-proceedings/data/html/papers/3178.pdf, doi:10.3182/20110828-6-IT-1002.03178.

[13] Hui-Min Huang, Kerry Pavek, James Albus, and Elena Messina. Autonomy levels for unmanned systems (ALFUS) framework: an update. *Proc. SPIE*, 5804:439–448, 2005. URL: http://dx.doi.org/10.1117/12.603725, doi:10.1117/12.603725.

[14] Bt Clough. Metrics, schmetrics! How the heck do you determine a UAV's autonomy anyway? *Security*, (990):313–319, 2002. URL: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA515926.

[15] LSTS. URL: http://lsts.fe.up.pt/.

[16] Sotiris Papatheodorou, Anthony Tzes, and Yiannis Stergiopoulos. Collaborative visual area coverage. *Robotics and Autonomous Systems*, 92:126–138, 2017. URL: http://dx.doi.org/10.1016/j.robot.2017.03.005, doi:10.1016/j.robot.2017.03.005.

[17] W.H. Huang. Optimal line-sweep-based decompositions for coverage algorithms. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 1, pages 27–32. IEEE, 2001. URL: http://ieeexplore.ieee.org/document/932525/, doi:10.1109/ROBOT.2001.932525.

[18] Gustavo S C Avellar, Guilherme A S Pereira, Luciano C A Pimenta, and Paulo Iscold. Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors (Switzerland)*, 15(11):27783–27803, 2015. doi:10.3390/s151127783.

[19] Yiannis Stergiopoulos and Anthony Tzes. Cooperative positioning/orientation control of mobile heterogeneous anisotropic sensor networks for area coverage. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1106–1111, 5 2014. URL: http://ieeexplore.ieee.org/document/6906992/, doi:10.1109/ICRA.2014.6906992.

[20] Suraj G. Gupta, Mangesh M. Ghonge, and P.M. Jawandhiya. Review of Unmanned Aircraft System. *International Journal of Advanced Research in Computer Engineering & Technology*, 2(4):2278–1323, 2013.

[21] Raj Jain and Fred L Templin. Wireless Datalink for Unmanned Aircraft Systems: Requirements, Challenges and Design Ideas. URL: http://www.cse.wustl.edu/~jain/papers/ftp/uas_dl.pdf.

[22] R John Hansman, Thesis Supervisor, and Jaime Peraire. Safety Considerations for Operation of Different Classes of Unmanned Aerial Vehicles in the National Airspace System by. 2005. URL: https://pdfs.semanticscholar.org/eb6f/d21018c507d9920ed2ca0ffe382bb441311d.pdf.

[23] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12):399, 2013. doi:10.5772/57313.

[24] F. Klauser and S. Pedrozo. Power and space in the drone age: a literature review and politico-geographical research agenda. *Geographica Helvetica*, 70(4):285–293, 10 2015. URL: http://www.geogr-helv.net/70/285/2015/, doi:10.5194/gh-70-285-2015.

[25] Eurocontrol. European ATM Master Plan: The Roadmap for Delivering High Performing Aviation for Europe. 2015. URL: https://ec.europa.eu/transport/sites/transport/files/modes/air/sesar/doc/eu-atm-master-plan-2015.pdf.

[26] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots*. Springer Japan, Tokyo, 2010. URL: http://link.springer.com/10.1007/978-4-431-53856-1, doi:10.1007/978-4-431-53856-1.

[27] Remote Piloted Aerial Vehicles: An Anthology, 2016. URL: http://www.ctie.monash.edu/hargrave/rpav_home.html#Beginningshttp://www.ctie.monash.edu.au/.

[28] Zain Anwar Ali, Dao-Bo Wang, M Kashif, Zaid Salim, and Faizan Hussain. Scientia Research Library UNMANNED AIRCRAFT SYSTEM. *Journal of Engineering And Technology Research*, 2(6), 2014. URL: http://www.scientiaresearchlibrary.

[29] Laurence R. Newcome. *Unmanned aviation: a brief history of unmanned aerial vehicles*. American Institute of Aeronautics and Astronautics, Reston ,VA, 1 2004. URL: http://arc.aiaa.org/doi/book/10.2514/4.868894http://books.google.com/books?id=2QFUAAAAMAAJ&pgis=1, doi:10.2514/4.868894.

[30] Dayton Wright/Kettering Liberty Eagle ("Bug"). URL: http://www.designation-systems.net/dusrm/app4/bug.html.

[31] de Havilland DH82B Queen Bee – de Havilland Aircraft Museum. URL: http://www.dehavillandmuseum.co.uk/aircraft/de-havilland-dh82b-queen-bee/.

[32] Warplanes: Russia Buys A Bunch Of Israeli UAVs. URL: https://www.strategypage.com/htmw/htairfo/articles/20090409.aspx.

[33] Henry H. Perritt and Eliot O. Sprague. *Domesticating drones : the technology, law and economics of unmanned aircraft*. Routledge. URL: https://www.routledge.com/Domesticating-Drones-The-Technology-Law-and-Economics-of-Unmanned-Aircraft/Perritt-Jr-Sprague/p/book/9781472458629.

[34] Ronald I. C. Bartsch, James Coyne, and Katherine Gray. *Drones in society : exploring the strange new world of unmanned aircraft*. Routledge. URL: https://www.routledge.com/Drones-in-Society-Exploring-the-strange-new-world-of-unmanned-aircraft/Bartsch-Coyne-Gray/p/book/9781138221574.

[35] Kristin Bergtora Sandvik and Maria Gabrielsen Jumbert. *The Good Drone*. Routledge, 2016. URL: https://www.routledge.com/The-Good-Drone/Sandvik-Jumbert/p/book/9781472451118papers3://publication/uuid/A95A44B4-211A-4753-85AE-1831291A2C58.

[36] Harrison G. Wolf. *Drones Safety Risk Management for the Next Evolution of Flight*. Taylor and Francis, 2017. URL: https://www.routledge.com/Drones-Safety-Risk-Management-for-the-Next-Evolution-of-Flight/Wolf/p/book/9781138203556.

[37] Ann Cavoukian. Privacy and Drones : Unmanned Aerial Vehicles. *Privacy by Design, Canada*, 1(1):1–27, 2012. URL: https://www.ipc.on.ca/wp-content/uploads/Resources/pbd-drones.pdf.

[38] N. Homainejad and C. Rizos. Application of multiple categories of unmanned aircraft systems (UAS) in different airspaces for bushfire monitoring and response. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(1W4):55–60, 2015. doi:10.5194/isprsarchives-XL-1-W4-55-2015.

[39] CAA. Unmanned Aircraft System Operations in UK Airspace – Guidance. *Cap 722*, (5):165, 2015. URL: www.caa.co.uk.

[40] Eren Turanoguz and Nafiz Alemdaroglu. Design of a medium range tactical UAV and improvement of its performance by using winglets. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1074–1083. IEEE, 6 2015. URL: http://ieeexplore.ieee.org/document/7152399/, doi:10.1109/ICUAS.2015.7152399.

[41] Emmanouil N. Barmpounakis and John C. Golias. Unmanned Aerial Aircraft Systems for transportation engineering: Current practice and future challenges. *International Journal of Transportation Science and Technology*, 5(3):111–122, 2016. URL: http://www.sciencedirect.com/science/article/pii/S2046043016300533, doi:10.1016/j.ijtst.2017.02.001.

[42] Guowei Cai, Jorge Dias, and Lakmal Seneviratne. A Survey of Small-Scale Unmanned Aerial Vehicles: Recent Advances and Future Development Trends. *Unmanned Systems*, 02(02):175–199, 4 2014. URL: http://www.worldscientific.com/doi/abs/10.1142/S2301385014300017, doi:10.1142/S2301385014300017.

[43] Daniel J Thompson, Joshuo Feys, Michael D Filewich, Sharif Abdel-magid, Dennis Dalli, and Fumitaka Goto. The Design and Construction of a Blended Wing Body UAV. *49th AIAA Aerospace Sciences Meeting*, (4-7 January):1–11, 2011.

[44] Robert Stengel. Sensors and Actuators! URL: http://www.princeton.edu/~stengel/MAE345.html.

[45] A.R. Jha. *Theory, Design, and Applications of Unmanned Aerial Vehicles - Books on Google Play*. CRC Press LLC, 2016. URL: https://play.google.com/store/books/details?id=guGVDQAAQBAJ&rdid=book-guGVDQAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport.

[46] W.H. Huang, Matthias R. Brust, Mustafa Ilhan Akbas, Damla Turgut, Andres Hernandez, Harold Murcia, Cosmin Copot, Joao Fortuna, Thor I. Fossen, Lateral Track, Control Law, F O R Aerosonde Uav, Joao Fortuna, Erik Johannes Forsmo, B. Moses Sathyaraj, L. C. Jain, A. Finn, S. Drake, Gustavo S C Avellar, Guilherme A S Pereira, Luciano C A Pimenta, Paulo Iscold, Sotiris Papatheodorou, Anthony Tzes, Yiannis Stergiopoulos, Matthew Turpin, Nathan Michael, Vijay Kumar, Markus Hehn, Raffaello D Andrea, Diogo Cardoso, Yiannis Stergiopoulos, Anthony Tzes, Mac Schwager, Brian J. Julian, Michael Angermann, Daniela Rus, Jose Joaquin Acevedo, Bego??a C. Arrue, Ivan Maza, Anibal Ollero, Kimon P. Valavanis, George J. Vachtsevanos, Sotiris Papatheodorou, Anthony Tzes, Yiannis Stergiopoulos, Ran Dai, John Cochran, Anastasios M. Lekkas, Thor I. Fossen, T. Espinoza, A. E. Dzul, R. Lozano, P. Parada, Timm Faulwasser, Janine Matschek, Pablo Zometa, Rolf Findeisen, F. Klauser, S. Pedrozo, Timm Faulwasser, Rolf Findeisen, Boaz Ben Moshe, Nir Shvalb, Jonathan Baadani, Itay Nagar, Harel Levy, Adam C. Watts, Vincent G. Ambrosia, Everett A. Hinkley, M Dempsey, Shahid Mahmood, Daniel J Thompson, Joshuo Feys, Michael D Filewich, Sharif Abdel-magid, Dennis Dalli, Fumitaka Goto, Suraj G. Gupta, Mangesh M. Ghonge, P.M. Jawandhiya, Aculdade De, Niversidade Do, Paulo Dias Ricardo Martins, Pedro Aguiar, Mehran Mesbahi, Magnus Egerstedt, a Noth, S Bouabdallah, R Siegwart, Krzysztof Kurek, Tomasz Keller, Jozef Modelski, Yevhen Yashchyshyn, Marcin Piasecki, Grzegorz Pastuszak, Marcin Darmetko, Pawel Bajurko, Hui-Min Huang, Kerry Pavek, James Albus, Elena Messina, Adrian Agogino, Chris HolmesParker, Kagan Tumer, Zhi Yan, Nicolas Jouandeau, Arab Ali Cherif, Alessandro Marino, Gianluca Antonelli, Antonio Pedro Aguiar, Antonio Pascoal, and Stefano Chiaverini. Collaborative visual area coverage. *Robotics and Autonomous Systems*, 92(1):126–138, 2012. URL: http://dl.acm.org/citation.cfm?id=2330163.2330306http://dx.doi.org/10.1117/12.603725http://www.transnav.eu/Article_Integrated_Communications_System_Kurek,26,432.htmlhttp://books.google.com/books?id=GlrqRwumdDcC&pgis=1%5Cnhttps://books.google.com/books?, doi:10.1016/j.robot.2017.03.005.

[47] Mary E Griswold. *Management Key to the Future of Unmanned Aircraft Systems ?* Number 44. 2008.

[48] Jose Joaquin Acevedo, Bego??a C. Arrue, Ivan Maza, and Anibal Ollero. Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints. *International Journal of Advanced Robotic Systems*, 10, 2013. doi:10.5772/52765.

[49] Shahid Mahmood. Unmanned Aerial Vehicle (UAV) Communications. *Power*, page 62, 2007.

[50] Esteban Gutiérrez Fernández. MASTER THESIS Management System for Unmanned Aircraft System Teams. *Technology*, (May), 2010. URL: http://upcommons.upc.edu/bitstream/handle/2099.1/9672/memoria.pdf?sequence=1.

[51] Reza Olfati-Saber. Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles. *Thesis PhD*, page 282, 2001.

[52] Robert F. Stengel. *Flight dynamics*. URL: https://books.google.pt/books/about/Flight_Dynamics.html?id=-OjZBQAAQBAJ&redir_esc=y.

[53] Andres Hernandez, Harold Murcia, and Cosmin Copot. Model Predictive Path-Following Control of an AR . Drone Quadrotor. pages 618–623, 2014.

[54] Joao Fortuna and Thor I. Fossen. Cascaded line-of-sight path-following and sliding mode controllers for fixed-wing UAVs. *2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings*, pages 798–803, 2015. doi:10.1109/CCA.2015.7320715.

[55] T. Espinoza, A. E. Dzul, R. Lozano, and P. Parada. Backstepping - Sliding Mode Controllers Applied to a Fixed-Wing UAV. *Journal of Intelligent & Robotic Systems*, 73(1-4):67–79, 1 2014. URL: http://link.springer.com/10.1007/s10846-013-9955-y, doi:10.1007/s10846-013-9955-y.

[56] Pedro Aguiar. Trajectory-tracking controllers for UAVs using DUNE. 2014.

[57] Lateral Track, Control Law, and F O R Aerosonde Uav. Aiaa 2001–0016. *Control*, (January), 2001.

[58] Talya Porat, Tal Oron-Gilad, Michal Rottem-Hovev, and Jacob Silbiger. Supervising and controlling unmanned systems: A multi-phase study with subject matter experts. *Frontiers in Psychology*, 7(MAY):568, 2016. URL: http://www.ncbi.nlm.nih.gov/pubmed/27252662http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4878290, doi:10.3389/fpsyg.2016.00568.

[59] Mary Cummings. Automation Bias in Intelligent Time Critical Decision Support Systems. *AIAA 1st Intelligent Systems Technical Conference*, (September):1–6, 2004. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.91.2634http://arc.aiaa.org/doi/pdf/10.2514/6.2004-6313%0Ahttp://arc.aiaa.org/doi/abs/10.2514/6.2004-6313%5Cnhttp://arc.aiaa.org/doi/pdf/10.2514/6.2004-6313, doi:doi:10.2514/6.2004-6313.

[60] Hui-Min Huang. Autonomy Levels for Unmanned Systems (ALFUS) Framework Volume I: Terminology Version 1.1 EXECUTIVE SUMMARY. *NIST Special Publication*, 1011. URL: https://www.nist.gov/sites/default/files/documents/el/isd/ks/NISTSP_1011_ver_1-1.pdf.

[61] AIR-VID. 20 great UAV applications areas for Drones, 2015. URL: https://air-vid.com/20-great-uav-applications-areas-drones/http://air-vid.com/wp/20-great-uav-applications-areas-drones/.

[62] Drone applications: Aerial Photography, Monitoring, Search & Rescue etc. URL: https://www.microdrones.com/en/applications/.

[63] Alessandro Marino, Gianluca Antonelli, Antonio Pedro Aguiar, Antonio Pascoal, and Stefano Chiaverini. A decentralized strategy for multirobot sampling/patrolling: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 23(1):313–322, 2015. doi:10.1109/TCST.2014.2312550.

[64] Peter Stone, Park Ave, and Florham Park. Multiagent Systems : A Survey from a Machine Learning Perspective. *Robotics*, 8(3):345–383, 1995. URL: www.research.att.www.cs.cmu.http://www.springerlink.com/index/w2671g4q27327305.pdf, doi:10.1023/A:1008942012299.

[65] Wolfram Burgard, Dieter Fox, Mark Moors, Reid Simmons, and Sebastian Thrun. Collaborative Multi-Robot Exploration. In *Proceedings of the {IEEE} International Conference on Robotics and Automation*, pages 476–481,

2000. URL: https://iths.pure.elsevier.com/en/publications/collaborative-multi-robot-exploration.

[66] Jens Wawerla and Richard T. Vaughan. A fast and frugal method for team task allocation in a multi-robot transportaton system. *Proc. of the Int. Conf. on Robotics and Automation*, pages 1432–1437, 2013. doi:10.1109/ROBOT.2010.5509865.

[67] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous robots*, 8(3):325–344, 2000.

[68] Stergios I Roumeliotis and George A Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[69] Raj Madhavan, Kingsley Fregene, and Lynne E Parker. Distributed heterogeneous outdoor multi-robot localization. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 374–381. IEEE, 2002.

[70] Alphan Ulusoy, Stephen L Smith, Xu Chu Ding, and Calin Belta. Robust multi-robot optimal path planning with temporal logic constraints. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4693–4698. IEEE, 2012.

[71] Stuart J Russell and Peter Norvig. Artificial intelligence: a modern approach (International Edition). 2002.

[72] Wei Ye, Richard T Vaughan, Gaurav S Sukhatme, John Heidemann, Deborah Estrin, and Maja J Mataric. Evaluating control strategies for wireless-networked robots using an integrated robot and network simulation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2941–2947. IEEE, 2001.

[73] nsnam web pages. URL: https://www.isi.edu/nsnam/.

[74] Paul E Rybski, Sascha A Stoeter, Maria Gini, Dean F Hougen, and Nikolaos P Papanikolopoulos. Performance of a distributed robotic system using shared communications channels. *IEEE transactions on Robotics and Automation*, 18(5):713–727, 2002.

[75] Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.

[76] Alessandro Farinelli, Luca Iocchi, and Daniele Nardi. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2015–2028, 2004.

[77] Jean-Claude Latombe, Anthony Lazanas, and Shashank Shekhar. Robot motion planning with uncertainty in control and sensing. *Artificial Intelligence*, 52(1):1–47, 1991.

[78] Yi Guo and Lynne E Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2612–2619. IEEE, 2002.

[79] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[80] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1):166–171, 1998.

[81] Gildardo Sanchez and J-C Latombe. Using a PRM planner to compare centralized and decoupled planning for multi-robot systems. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 2112–2119. IEEE, 2002.

[82] Mitul Saha and Pekka Isto. Multi-robot motion planning by incremental coordination. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5960–5963. IEEE, 2006.

[83] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[84] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics and Automation Magazine*, 19(1):24–39, 3 2012. URL: http://ieeexplore.ieee.org/document/6161683/, doi:10.1109/MRA.2011.2181683.

[85] Sonia Waharte and Niki Trigoni. Supporting search and rescue operations with UAVs. In *Proceedings - EST 2010 - 2010 International Conference on Emerging Security Technologies, ROBOSEC 2010 - Robots and Security, LAB-RS 2010 - Learning and Adaptive Behavior in Robotic Systems*, number September 2010, pages 142–147, 2010. URL: https://www.cs.ox.ac.uk/files/3198/submission_waharte.pdf, doi:10.1109/EST.2010.31.

[86] Francesco Nex and Fabio Remondino. UAV for 3D mapping applications: A review, 3 2014. URL: http://link.springer.com/10.1007/s12518-013-0120-x, doi:10.1007/s12518-013-0120-x.

[87] Antonio Barrientos, Julian Colorado, Jaime del Cerro, Alexander Martinez, Claudio Rossi, David Sanz, and João Valente. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5):667–689, 9 2011. URL: http://doi.wiley.com/10.1002/rob.20403, doi:10.1002/rob.20403.

[88] Pavichaya Eaungpulswat. *Area Coverage Algorithms for Multiagent Surveillance Tasks*. PhD thesis, Technische Universität Hamburg-Harburg, 2012. URL: https://www.ifis.uni-luebeck.de/~moeller/publist-sts-pw-and-m/source/papers/2012/eaung12.pdf.

[89] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. 2010. URL: http://books.google.com/books?id=GlrqRwumdDcC&pgis=1%5Cnhttps://books.google.com/books?hl=en&lr=&id=GlrqRwumdDcC&oi=fnd&pg=PP2&dq=graph+theoretic+methods+in+multiagent+networks&ots=e2RXMVsBgh&sig=_8xWEqluMAk6J5UTZwAJNoorlo0, doi:10.1073/pnas.0703993104.

[90] Sotiris Papatheodorou, Anthony Tzes, and Yiannis Stergiopoulos. Collaborative visual area coverage. *Robotics and Autonomous Systems*, 92:126–138, 2017. URL: http://dx.doi.org/10.1016/j.robot.2017.03.005, doi:10.1016/j.robot.2017.03.005.

[91] Chao Gui and Prasant Mohapatra. Virtual Patrol: A New Power Conservation Design for Surveillance Using Sensor Networks. *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 511,

2005. URL: http://delivery.acm.org/10.1145/1150000/1147728/a33-gui.pdf?ip=193.136.33.132&id=1147728&acc=ACTIVESERVICE&key=2E5699D25B4FE09E.6F699AA92A518455.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=949370607&CFTOKEN=15793904&__acm__=1497781899_d577eda281ac175fc701c790.

[92] Yan Li, Hai Chen, Meng Joo Er, and Xinmin Wang. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, 21(5):876–885, 8 2011. URL: http://linkinghub.elsevier.com/retrieve/pii/S0957415810001893, doi:10.1016/j.mechatronics.2010.10.009.

[93] Spencer G Fowers. Stabilization and Control of a Quad-Rotor Micro- UAV Using Vision Sensors. 2008. URL: http://scholarsarchive.byu.edu/etd.

[94] Nojeong Heo and P.K. Varshney. Energy-efficient deployment of Intelligent Mobile sensor networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):78 – 92, 1 2005. URL: http://ieeexplore.ieee.org/document/1369347/http://ieeexplore.ieee.org/ielx5/3468/29969/01369347.pdf?tp=&arnumber=1369347&isnumber=29969%5Cnhttp://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1369347, doi:10.1109/TSMCA.2004.838486.

[95] Y. Bouktir, M. Haddad, and T. Chettibi. Trajectory planning for a quadrotor helicopter. *2008 16th Mediterranean Conference on Control and Automation*, pages 1258–1263, 6 2008. URL: http://ieeexplore.ieee.org/document/4602025/, doi:10.1109/MED.2008.4602025.

[96] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 12 2013. URL: http://linkinghub.elsevier.com/retrieve/pii/S092188901300167X, doi:10.1016/j.robot.2013.09.004.

[97] Howie Choset. Coverage of Known Spaces: The Boustrophedon Cellular Decomposition. *Autonomous Robots*, 9(3):247–253, 2000. URL: http://link.springer.com/10.1023/A:1008958800904, doi:A:1008958800904.

[98] Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, and Douglas Hull. Morse Decompositions for Coverage Tasks. *The International Journal of Robotics Research*, 21(4):331–344, 4 2002. URL: http://ijr.sagepub.com/cgi/doi/10.1177/027836402320556359http://ijr.sagepub.com/content/21/4/331.abstract, doi:10.1177/027836402320556359.

[99] Ercan U. Acar, Howie Choset, and Ji Yeong Lee. Sensor-based coverage with extended range detectors. *IEEE Transactions on Robotics*, 22(1):189–198, 2 2006. URL: http://ieeexplore.ieee.org/document/1589013/, doi:10.1109/TRO.2005.861455.

[100] Ivan Maza and Anibal Ollero. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In *Distributed Autonomous Robotic Systems 6*, pages 221–230. Springer Japan, Tokyo, 2007. URL: http://www.springerlink.com/index/10.1007/978-4-431-35873-2_22, doi:10.1007/978-4-431-35873-2{_}22.

[101] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. *Autonomous Robots*, 36(4):365–381, 4 2014. URL: http://link.springer.com/10.1007/s10514-013-9364-x, doi:10.1007/s10514-013-9364-x.

[102] Jean-François Cordeau, Gilbert Laporte, Martin W P Savelsbergh, and Daniele Vigo. Chapter 6 Vehicle Routing. *Transportation*, 14(06):367–428, 2007. URL: http://dis.unal.edu.co/~gjhernandezp/TOS/ROUTING/VRP1.pdf, doi:http://dx.doi.org/10.1016/S0927-0507(06)14006-2.

[103] M. Alighanbari, Y. Kuwata, and J.P. How. Coordination and control of multiple UAVs with timing constraints and loitering. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 6, pages 5311–5316. IEEE. URL: http://ieeexplore.ieee.org/document/1242572/, doi:10.1109/ACC.2003.1242572.

[104] Kjerstin Easton and Joel Burdick. A coverage algorithm for multi-robot boundary inspection. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2005, pages 727–734. IEEE, 2005. URL: http://ieeexplore.ieee.org/document/1570204/, doi:10.1109/ROBOT.2005.1570204.

[105] Halit Ergezer and Kemal Leblebicio. 3D Path Planning for Multiple UAVs for Maximum. *J Intell Robot Syst (2014)*, 2014(73):737–762, 1 2014. URL: http://link.springer.com/10.1007/s10846-013-9895-6, doi:10.1007/s10846-013-9895-6.

[106] F. Guerriero, R. Surace, V. Loscr??, and E. Natalizio. A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints. *Applied Mathematical Modelling*, 38(3):839–852, 2 2014. URL: http://linkinghub.elsevier.com/retrieve/pii/S0307904X13004460, doi:10.1016/j.apm.2013.07.002.

[107] Mesut. Gunes. *Modellierung und Bewertung von Kommunikationssystemen : Skript zur Vorlesung an der RWTH Aachen*. Mainz, 2001. URL: http://cst-pub.imp.fu-berlin.de/Spaniol2001.txt.

[108] Von der Carl-Friedrich Gauß-Fakultät, Von Habib-ur Rehman, Dr-Ing Lars Wolf Korreferent, and Utz Roedig. Multihop Connectivity in Wireless Ad hoc Networks. 2009. URL: https://publikationsserver.tu-braunschweig.de/servlets/MCRFileNodeServlet/digibib_derivate_00006192/Dissertation.pdf.

[109] Dajin Wang, Li Xu, Jing Peng, and Stefan Robila. Subdividing hexagon-clustered wireless sensor networks for power-efficiency. In *Proceedings - 2009 WRI International Conference on Communications and Mobile Computing, CMC 2009*, volume 2, pages 454–458. IEEE, 1 2009. URL: http://ieeexplore.ieee.org/document/4797165/, doi:10.1109/CMC.2009.317.

[110] N. Heo and P K Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. *2003 IEEE Wireless Communications and Networking 2003 WCNC 2003*, 3(C):1597–1602, 2003. URL: http://ieeexplore.ieee.org/document/1200625/http://www.springerlink.com/content/r226v11248540521/, doi:10.1109/WCNC.2003.1200625.

[111] Chan-myung Kim, Yong-hwan Kim, Hee-Sung Lim, and Youn-Hee Han. A Self-deployment Scheme for Mobile Sensor Network with Obstacle Avoidance. pages 345–354, 2010. URL: http://link.springer.com/10.1007/978-3-642-16444-6_44, doi:10.1007/978-3-642-16444-6{_}44.

[112] Jorge Cortes, Sonia Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(4):691–719, 10 2004. URL: http://www.esaim-cocv.org/10.1051/cocv:2005024http://arxiv.org/abs/math/0401297, doi:10.1051/cocv.

[113] Luciano C A Pimenta, Vijay Kumar, Renato C Mesquita, and Guilherme A S Pereira. Sensing and coverage for a network of heterogeneous robots. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3947–3952, 2008. URL: https://pdfs.semanticscholar.org/cad1/624313e26c6b567c5c30671852d2f5a343b8.pdf, doi:10.1109/CDC.2008.4739194.

[114] Jorge Cortes, Sonia Martinez, Timur Karata??, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 4 2004. URL: http://ieeexplore.ieee.org/document/1284411/, doi:10.1109/TRA.2004.824698.

[115] A. Tzes and Y. Stergiopoulos. Convex Voronoi-inspired space partitioning for heterogeneous networks: a coverage-oriented approach. *IET Control Theory & Applications*, 4(12):2802–2812, 12 2010. URL: http://digital-library.theiet.org/content/journals/10.1049/iet-cta.2009.0298, doi:10.1049/iet-cta.2009.0298.

[116] Omur Arslan and Daniel E. Koditschek. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2016-June, pages 4259–4266. IEEE, 5 2016. URL: http://ieeexplore.ieee.org/document/7487622/, doi:10.1109/ICRA.2016.7487622.

[117] Minh Tri Nguyen, Luis Rodrigues, Cristina Stoica Maniu, and Sorin Olaru. Discretized optimal control approach for dynamic multi-agent decentralized coverage. In *IEEE International Symposium on Intelligent Control - Proceedings*, volume 2016-Septe, pages 1–6. IEEE, 9 2016. URL: http://ieeexplore.ieee.org/document/7579984/, doi:10.1109/ISIC.2016.7579984.

[118] Minh Tri Nguyen and Cristina Stoica Maniu. Voronoi based decentralized coverage problem: From optimal control to Model Predictive Control. In *Proc. 24th Mediterranean Conf. Control and Automation (MED)*, pages 1307–1312. IEEE, 6 2016. URL: http://ieeexplore.ieee.org/document/7535958/, doi:10.1109/MED.2016.7535958.

[119] Fatemeh Mohseni, Ali Doustmohammadi, and Mohammad Bagher Menhaj. Distributed Receding Horizon Coverage Control for Multiple Mobile Robots. *IEEE Systems Journal*, 10(1):198–207, 3 2016. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6828743, doi:10.1109/JSYST.2014.2325219.

[120] Vinod Ramaswamy and Jason R. Marden. A sensor coverage game with improved efficiency guarantees. In *Proceedings of the American Control Conference*, volume 2016-July, pages 6399–6404. IEEE, 7 2016. URL: http://ieeexplore.ieee.org/document/7526676/, doi:10.1109/ACC.2016.7526676.

[121] Yiannis Stergiopoulos and Anthony Tzes. Spatially distributed area coverage optimisation in mobile robotic networks with arbitrary convex anisotropic patterns. *Automatica*, 49(1):232–237, 1 2013. URL: http://linkinghub.elsevier.com/retrieve/pii/S0005109812004736, doi:10.1016/j.automatica.2012.09.012.

[122] Yiannis Kantaros, Michalis Thanou, and Anthony Tzes. Distributed coverage control for concave areas by a heterogeneous Robot-Swarm with visibility sensing constraints. *Automatica*, 53:195–207, 3 2015. URL: http://linkinghub.elsevier.com/retrieve/pii/S000510981400613X, doi:10.1016/j.automatica.2014.12.034.

[123] Dimitra Panagou, Dusan M. Stipanovic, and Petros G. Voulgaris. Distributed dynamic coverage and avoidance control under anisotropic sensing. *IEEE Transactions on Control of Network Systems*, pages 1–1, 2016. URL: http://ieeexplore.ieee.org/document/7484288/, doi:10.1109/TCNS.2016.2576403.

[124] E. Bakolas. Distributed partitioning algorithms for multi-agent networks with quadratic proximity metrics and sensing constraints. *Systems and Control Letters*, 91:36–42, 5 2016. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167691116000347, doi:10.1016/j.sysconle.2016.02.005.

[125] Yiannis Stergiopoulos, Michalis Thanou, and Anthony Tzes. Distributed Collaborative Coverage-Control Schemes for Non-Convex Domains. *IEEE Transactions on Automatic Control*, 60(9):2422–2427, 9 2015. URL: http://ieeexplore.ieee.org/document/7054473/, doi:10.1109/TAC.2015.2409903.

[126] Reza Javanmard Alitappeh, Kossar Jeddisaravi, and Frederico G. Guimarães. Multi-objective multi-robot deployment in a dynamic environment. *Soft Computing*, pages 1–17, 6 2016. URL: http://link.springer.com/10.1007/s00500-016-2207-x, doi:10.1007/s00500-016-2207-x.

[127] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos. Multi-robot three-dimensional coverage of unknown areas. *The International Journal of Robotics Research*, 31(6):738–752, 5 2012. URL: http://ijr.sagepub.com/cgi/doi/10.1177/0278364912439332, doi:10.1177/0278364912439332.

[128] Andreas Breitenmoser, Jean-Claude Metzger, Roland Siegwart, and Daniela Rus. Distributed coverage control on surfaces in 3D space. *23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010, October 18, 2010 - October 22, 2010*, pages 5569–5576, 10 2010. URL: http://ieeexplore.ieee.org/document/5652851/http://dx.doi.org/10.1109/IROS.2010.5652851, doi:10.1109/IROS.2010.5652851.

[129] Michalis Thanou and Anthony Tzes. Distributed visibility-based coverage using a swarm of UAVs in known 3D-terrains. In *ISCCSP 2014 - 2014 6th International Symposium on Communications, Control and Signal Processing, Proceedings*, pages 425–428. IEEE,

5 2014. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6877904, doi:10.1109/ISCCSP.2014.6877904.

[130] Marina Torres, David A. Pelta, José L. Verdegay, and Juan C. Torres. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Systems with Applications*, 55:441–451, 8 2016. URL: http://linkinghub.elsevier.com/retrieve/pii/S0957417416300306, doi:10.1016/j.eswa.2016.02.007.

[131] Mac Schwager, Brian J. Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. In *Proceedings of the IEEE*, volume 99, pages 1541–1561, 9 2011. URL: http://ieeexplore.ieee.org/document/5959179/, doi:10.1109/JPROC.2011.2158377.

[132] J Fortuna. Mixed Initiative planning and control of UAV teams for Persistent Surveillance Final Preparation Report. 2012. URL: http://paginas.fe.up.pt/~ee07245/wp-content/uploads/2012/02/ee072451.pdf.

[133] Carmelo Di Franco and Giorgio Buttazzo. Coverage Path Planning for UAVs Photogrammetry with Energy and Resolution Constraints. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 83(3-4):445–462, 9 2016. URL: http://link.springer.com/10.1007/s10846-016-0348-x, doi:10.1007/s10846-016-0348-x.

[134] N. Christofides, A. Mingozzi, and P. Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282, 12 1981. URL: http://link.springer.com/10.1007/BF01589353, doi:10.1007/BF01589353.